COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2014; **25**:421–433
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cav.1594

SPECIAL ISSUE PAPER

Real-time physical deformation and cutting of heterogeneous objects via hybrid coupling of meshless approach and finite element method

Chen Yang[1], Shuai Li[1]*, Lili Wang[1], Aimin Hao[1] and Hong Qin[2]

[1] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, China
[3] Stony Brook University, USA

ABSTRACT

This paper advocates a method for real-time physical deformation and arbitrary cutting simulation of heterogeneous objects with multi-material distribution, whose originality centers on the tight coupling of domain-specific finite element method (FEM) and material distance-aware meshless approach in a CUDA-centric parallel simulation framework. We employ hierarchical hexahedron serving as basic building blocks for accurate material-aware FEM simulation. Meanwhile, local meshless systems are designed to support cross-FEM-domain coupling and material-sensitive propagation while respecting the regularity of finite elements. Directly benefiting from the structural regularity and uniformity of finite elements, our hybrid solution enables the local stiffness matrix pre-computation and dynamic assembling, adaptive topological updating and precise cutting reconstruction. Moreover, our mathematically-rigorous solver guarantees unconditional stableness. Experiments demonstrate the superiorities of our system. Copyright © 2014 John Wiley & Sons, Ltd.

KEYWORDS

physical simulation; heterogeneous objects; FEM; meshless coupling

***Correspondence**

Shuai Li, State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, China.
E-mail: ls@vrlab.buaa.edu.cn

1. INTRODUCTION

The physical deformation/cutting simulation has been gaining research interests in recent years. Despite the growing success, certain difficulties still prevail for high-fidelity simulation of heterogeneous objects. Key technical challenges include material-sensitive physical modeling, cross-material coupling, adaptive material-structure updating, and sophisticated computational schemes, which are summarized as follows.

First, most of the popular finite element method (FEM) methods discretize objects' physical domain using homogeneous elements. However, it is extremely difficult to model multi-scale heterogeneous object with a single FEM system supporting the same type/scale of finite elements because of complex handling of cross-material regions. Meanwhile, the dynamic updates of material-aware structures during cutting simulation together with the dynamic stiffness matrix assembly unavoidably give rise to heavy computational costs.

Second, even though meshless methods have natural advantages in accommodating topological changes, when confronting arbitrary cutting simulation of heterogeneous materials, its computational cost will drastically increase because of time-consuming update of global material distance field. Thus, there remains a large gap between the meshless flexibility and high FEM accuracy supporting heterogeneous materials.

Third, when focusing on the efficiency and stableness of numerical solvers, explicit GPU-based solvers indeed offer efficiency at the cost of being only conditionally stable. However, it may be too sensitive to the element structure, which is not suitable for material-aware strain kinematic formulations. Therefore, to accommodate stable material-aware simulation, CUDA-based domain-parallel implicit solvers are urgently needed.

As illustrated in Figure 1, we present a novel method to tackle the aforementioned challenges. Specifically, the salient contributions of this paper include the following:

- We systematically articulate a hybrid framework to interlink local material-specific FEMs with meshless blending systems. It facilitates the local stiffness

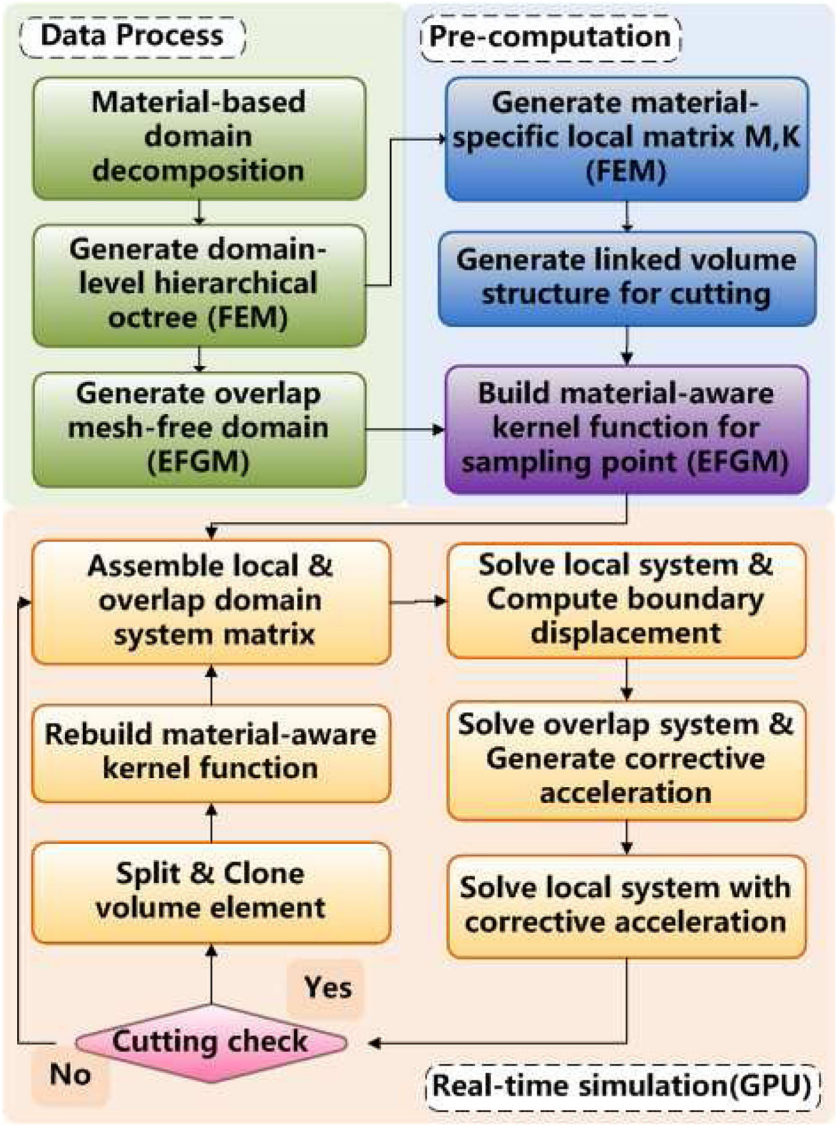


Figure 1. The flowchart of our method.

matrix pre-computation, dynamic topological updating and cutting surface reconstruction, and it can globally propagate the physical behaviors of local heterogeneous materials during deformation and arbitrary cutting simulation.

- We develop a dynamic material distance computing method for the domain-interlinking regions, which facilitates both the anisotropic meshless kernel function calculation and the cross-domain displacement field smoothing.
- We design a structure-free algorithm based on CUDA and devise a domain-parallel implicit solver for numerical integration, which can achieve real-time performance and stable physical simulation even for heterogeneous objects.

2. RELATED WORK

Finite element methods. The central idea of finite element methods is to discretize the physical domain using tetrahedral [1,2] or hexahedral [3] elements. Most of real-time FEM approaches are based on linear formulations. For example, Dick *et al.* [4] proposed a hexahedra-based FEM method by introducing a multi-grid solver on GPU. To enable nonlinear large deformation, Dick [5] extended the co-rotated strain formulation to handle hexahedral elements and achieved linear time complexity. Meanwhile, the Total Lagrangian explicit dynamics method proposed by Miller *et al.* rigorously formulates the non-linear constitutive law [6] and is further extended to handle anisotropic viscoelastic deformation [7]; however,

DOI: 10.1002/cav

it is conditionally stable. As for FEM-based cutting simulation, it remains challenging in dynamic topological update and realistic cutting surface reconstruction [8]. Finite element subdivision-based method [9] is usually employed to handle topological update, but it can create severely ill-conditioned simulation elements. Although Wicked *et al.* [10] improved it by using arbitrarily-convex finite elements, it is even more involved in numerical integration. Most recently, adaptive regular hexahedron approximation is used to cutting simulations [4]; however, it can only handle homogeneous objects.

Meanwhile, FEM model is often solved by employing either explicit method [7,11] or implicit integration scheme [4]. To improve this, Chaos *et al.* [12] proposed a combination of a geometric material model with a fully variational geometric integrator, and Fieri *et al.* [13] showed how to identify ill-shaped elements hindering stable numerical time integration. Moreover, based on element-wise stability considerations, a hybrid methodology combining explicit and implicit linear integration approaches is also proposed in [14].

Meshless methods. In sharp contrast to FEM, meshless approaches do not require any explicit topological structures [15,16], wherein physical behaviors are evaluated with material distance-based radial functions. Taking fracture for example, the material distance within the object should be dynamically updated in local regions [17], and Steinmann, Toady, and Gross [18] incorporate the connectivity information into particle-based meshless simulation. Although such methods can offer great flexibility in adaptive re-sampling, the advantage of meshless methods usually depends on the complex definitions of the shape functions. Thus, sparse frame-based models are introduced by Grilles [19], and then Fauve *et al.* [16] show that it is possible to simulate complex heterogeneous objects with material-aware shape functions-based sparse sampling.

Coupling methods. The idea of coupling Element-free Gherkin (EFT) and FE methods opens up a new venue to explore. Bellyache *et al.* [20] proposed a method to blend the EFT domain and FE domain by adding a layer of interface elements. And Rack *et al.* [21] gave a general overview of the coupling of meshless methods with finite elements.

Besides, multi-domain subspace techniques provide a flexible solution to reduce the physical model by partitioning the deformable object into multiple domains. Kim and James [22] proposed to couple the domains using penalty forces for character skinning. However, this method requires pre-determined motions of local frames for each domain. Another multi-domain subspace deformation method in [23] relies on shape matching and mass lumping at the boundary interfaces to handle the coupling issue. Most recently, Yang *et al.* [24] propose a boundary-aware mode construction method to characterize the deformation subspace of each domain, and Busman *et al.* [25] enable mechanical coupling and propagation of boundary conditions among domains by simulating the interface with six-degrees of freedoms (Doffs) mechanical points.

3. METHOD OVERVIEW

As shown in Figure 1, we first focus on the method overview and its functionalities, which integrate heterogeneous material handling and dynamic topological change into a new framework by tightly coupling domain-decomposed FEM and Meshless method.

Material-aware domain decomposition and interlink. We conduct material distribution analysis and decompose the already-labeled volume into independent homogeneous sub-domains (Figure 2(a)), and construct octroi-based domain-specific FEM structure (Figure 2(b and c)). Then we build the material-aware *Overlapping Meshless Domain* using the boundary element of the *Local Finite Element Domains* (Figure 3). In Figure 3, local finite element domain (cyan and red lattices) are modeled by FEM; two distinct colors represent different materials. Overlapping meshless domain (crosshatch) is modeled by Element-free Gherkin method (EFGM) composing of different material elements. The vertices of elements simultaneously serve as the octroi vertices in finite element system and the sampling points in meshless system. By forcing the boundary element to be in the finest level, the *Overlapping Meshless Domain* is built by the unified Material-aware background grid.

Domain-specific FEM modeling. For each homogeneous FEM domain, we element-wisely compute local matrix for Lagrangian equation with the material-specific element parameters. Benefitting from the structural regularity and uniformity of finite elements, we can pre-store the local stiffness matrix on GPU and look them up by domain ID in runtime.

Material-aware EFGM modeling. For each meshless domain, we generate assistant background grid and calculate the interlinking relationship of the sampling point set. Because the meshless domain is composed of different material elements, we compute the distance field of sampling point in its influence domain one by one [26]. Figure 2(d) is the construction procedure of material-aware distance field illustrated in 2D example. Figure 2(e) illustrates the accompanying material-aware kernel function. The aforementioned operation are all executed in parallel on the GPU for speed up. For example, Figure 4 shows the simulation comparison of the homogeneous bar and heterogenous bar (material ratio 6:1:3).

Cross-material domain coupling. During each simulation loop, each local finite element system is solved independently. According to the solution, we can obtain the displacement field of each boundary element. Then from that, the EFT system in overlapping domain can generate the material-aware displacement correction result. Thus, we could predict the correct acceleration of each local domain that would be applied to the implicit integration scheme.

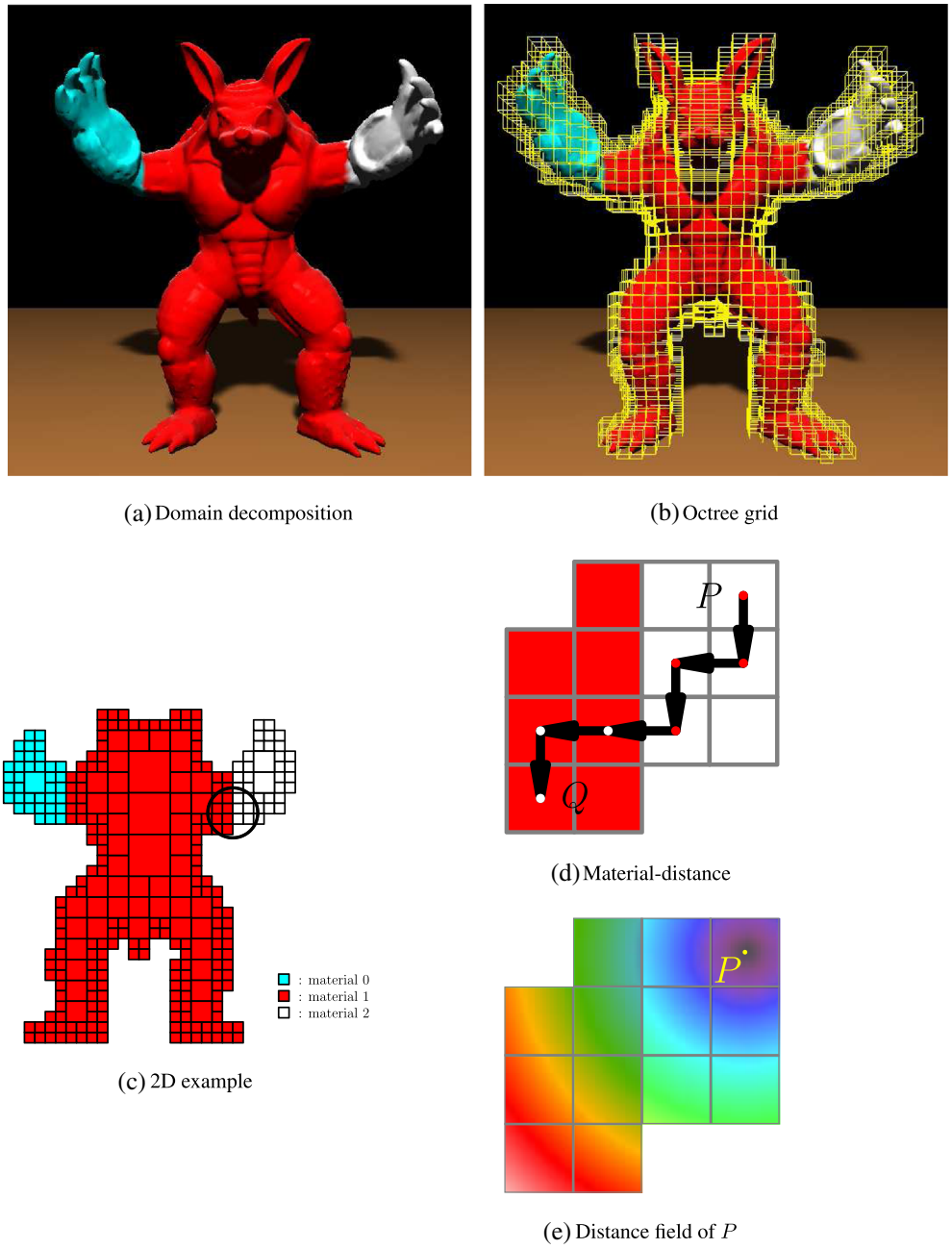


Figure 2. Octroi-based structure construction.

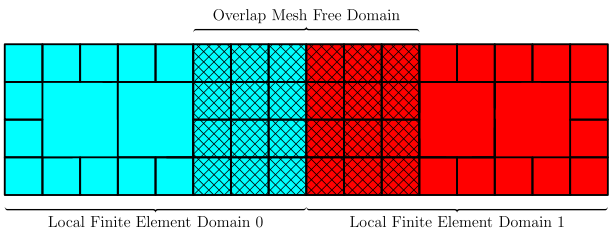
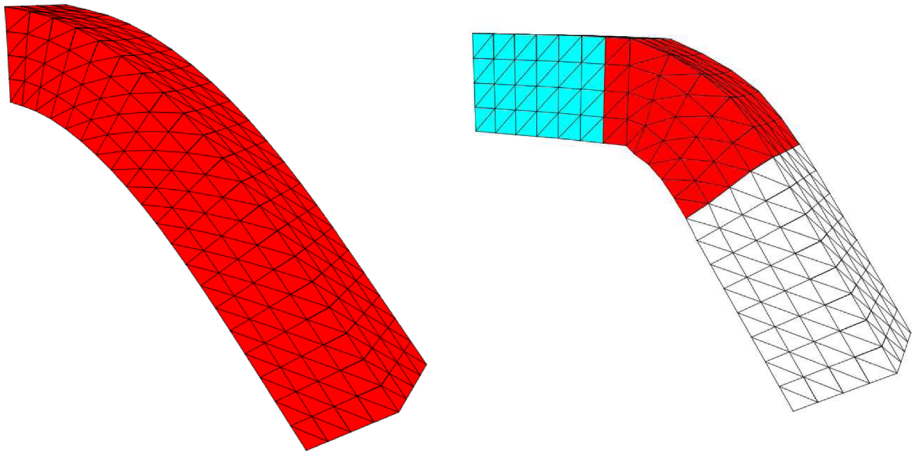


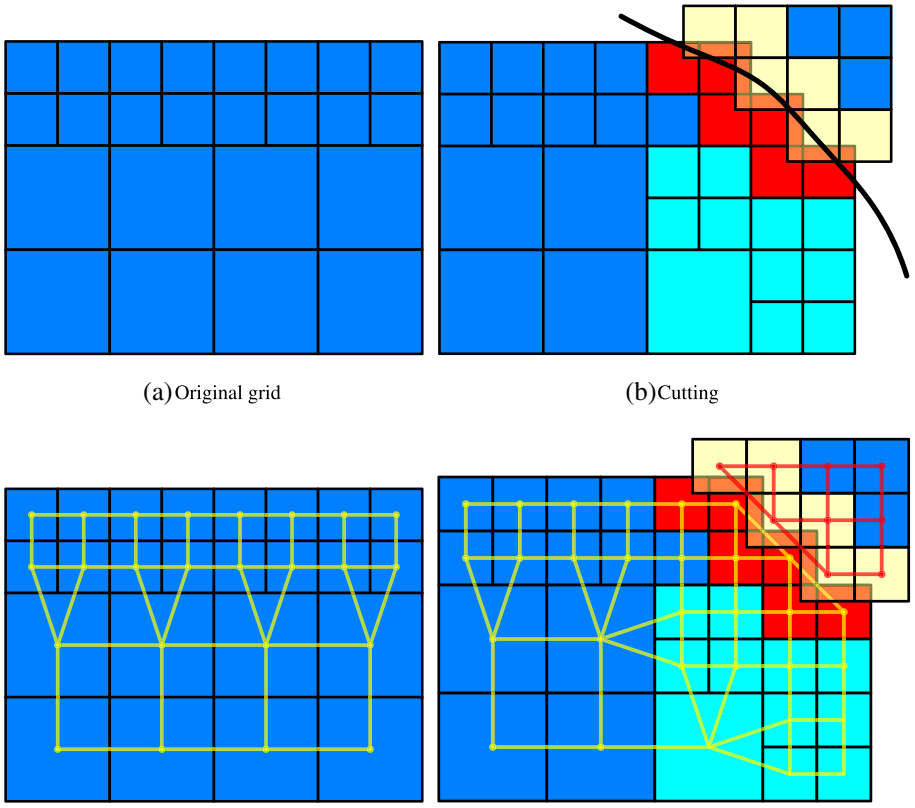
Figure 3. Meshless domain construction.

Comp. Anim. Virtual Worlds 2014; **25**:421–433 © 2014 John Wiley & Sons, Ltd.
DOI: 10.1002/cav



(a) Single material (b) Three materials

Figure 4. Deformation of heterogeneous bar.



(a) Original grid (b) Cutting

(c) Topological map (d) Updated topological map

Figure 5. Illustration of cutting modeling.

Cutting simulation. For geometrical structure updating, we employ element clone method to handle cutting action (Figure 5). When cutting, we refine the element intersecting with cutting blade (the cyan elements in Figure 5(b)), and clone the elements in red to form the new yellow elements. For meshless system updating that arises from material distance change, we renew topological information of cloned element (Figure 5(d)) and generate new distance field, which will be detailed in Section 4.4.

4. PHYSICAL SIMULATION

The dynamic behavior of a deformable object could be described by the Lagrangian equation:

$$M\ddot{u} + C\dot{u} + Ku = f \tag{1}$$

where M, C, and K, respectively, denote the mass, damping, and stiffness matrix of specific domain. The stiffness matrix and mass matrix of each element can be obtained by

$$\mathbf{k}_e = \int_{\Omega_e} B^T C B d\Omega \tag{2}$$

$$\mathbf{m}_e = \int_{\Omega_e} \rho N^T N d\Omega \tag{3}$$

C is a constant matrix related to material properties, while B and N, respectively, represent strain matrix and shape function matrix. Thus, the nodal force vector can be obtained through $\mathbf{f}_e = \int_{\Gamma_e} N^T f d\Gamma$. With the commonly adopted Rayleigh damping assumption, we can further define the damping matrix as $\mathbf{C} = \alpha M + \beta K$, where α and β are weighting coefficients.

For FEM-based sub-domains, we divide them into hexahedra. The displacement vector field $\mathbf{u}$ within each element can be approximated via shape functions comprising hexahedral vertex coordinates (x, y, z), and please refer to [27] for more detailed discussions.

4.1. Material-Aware Kernel Function

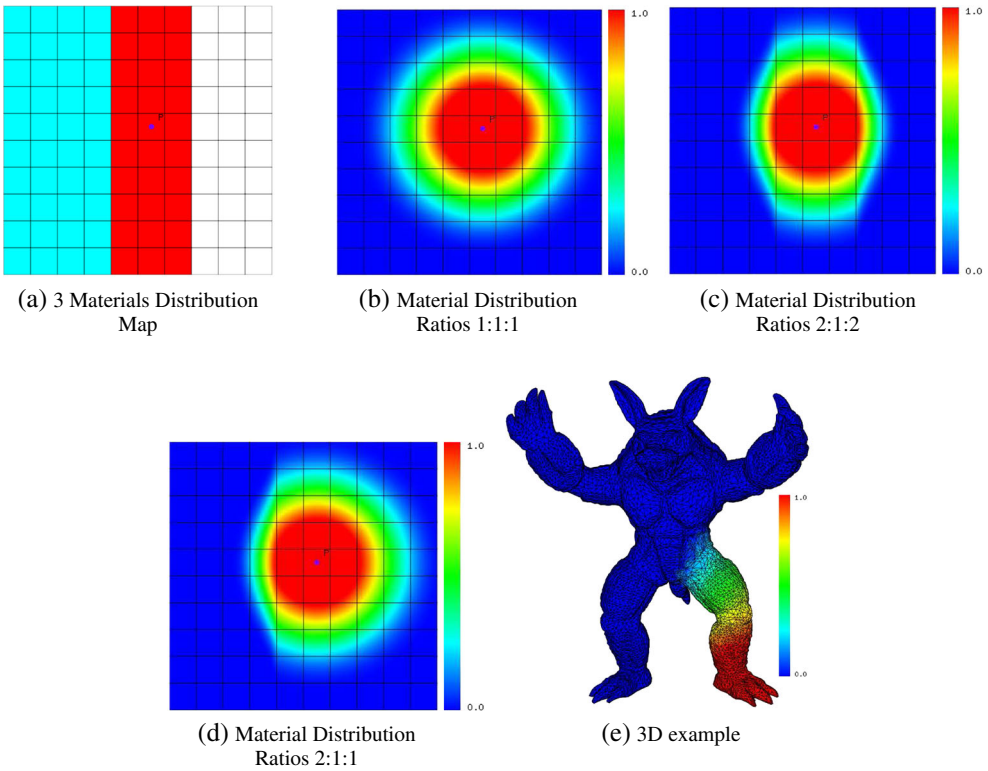
As for overlapping meshless domains, we take the hexahedral element as the assistant background grid to facilitate numerical integration. And the displacement $u(x)$ in the EFT-based domain Ω is interpolated via moving least squares (MLS) approximations with circle supporting region $u^h(x) = \sum_i w(r_x)u_i$ (the purple part of Figure 1). And the kernel function $w(r_x)$ is defined based on the material distance r_x between the sample point and the integration point x. Here, we employ a cubic spline function (Equation (4)) to serve as the basis function of MLS.

$$w(r) = \begin{cases} 2/3 - 4r^2 + 4r^3 & r \le 1/2 \\ 4/3 - 4r + 4r^2 - 4r^3/3 & 1/2 < r \le 1 \\ 0 & r > 1 \end{cases} \tag{4}$$

Inspired by Dionne and de Lasa [26], we reform the distance function between two adjoined elements (Equation (5)) to support multi-material case.

$$D_{material} = \left(\frac{m0 + m1}{2}\right) \times D_{euclid} \tag{5}$$

where $m0$ and $m1$ are material parameters. By resorting to the Dijkstra's algorithm in Equation (5), we could generate the material-aware distance field and subsequently obtain the material-aware kernel function for EFGM (Figure 6). Figure 6 intuitively show our material-aware kernel function corresponding to different material distribution ratios.



(a) 3 Materials Distribution Map (b) Material Distribution Ratios 1:1:1 (c) Material Distribution Ratios 2:1:2

(d) Material Distribution Ratios 2:1:1 (e) 3D example

Figure 6. Material-aware kernel functions.

4.2. Cross-Material Domain Coupling

To model the cross-material domain coupling, we develop a material-aware displacement correction method. As is mentioned previously, we firstly solve each local domain independently and obtain the displacement u_e^{fe} for each boundary element, which serves as the boundary condition applied to the meshless system in Equation (6).

$$u^{efg} = {\Phi_e}^T N_e u_e^{fe} \tag{6}$$

Here, N_e is the FEM shape function matrix, and Φ_e is the corresponding EFGM shape function matrix. By solving the EFT system, we can obtain the material-aware correction displacement field. And then we compute the correction acceleration and attach it to corresponding sub-domains. Once again, we solve each sub-domain independently as the final solution(see the real-time simulation part of Figure 1).

4.3. Modal Warping

In this paper, we adopt the most commonly used Cauchy strain tensor. However, because the linear Cauchy tensor tends to diverge from the correct solution under the large deformation, we build our simulation framework upon the co-rotational formulation. Figure 7 shows a comparison between linear strain and linear co-rational strain. In practice, we employ the warped stiffness approach [28] to explicitly compute and warp the local rotations of each element. Therefore, our approach can efficiently handle non-linear large-scale deformation.

$$\begin{aligned} &\textstyle\sum_{j=1}^{8} RK_{ij}R^T u_j = f_i - \sum_{j=1}^{8} RK_{ij}\left(R^T p_j^0 - p_j^0\right), \\ &i = 1, \ldots, 8 \end{aligned} \tag{7}$$

For the sake of parallel computation, we calculate the element's local rotation via the polar decomposition of the elements average deformation gradient. In each simulation cycle, we first compute the element's local rotation according to Equation (8). And we further construct the warp stiffness matrix using Equation (7) when assembling the system equations. Here, p_j^0 denotes the vertex position in the reference configuration.

$$R_i = \begin{cases} I_3 + \frac{1}{4}\sum_i^8 u_i^{old}\left(\pm\frac{1}{r}, \pm\frac{1}{r}, \pm\frac{1}{r}\right), & i = 0 \\ \frac{1}{2}\left(R_{i-1} + \left(R_{i-1}^{-1}\right)^T\right), & i > 0 \end{cases} \tag{8}$$

4.4. Cutting Simulation

To accommodate dynamic topological changes and accompanying update of the system equations, we detect the coarser-level elements that intersect with the cutting blade in each cutting simulation cycle. Next, they are adaptively refined to guarantee that there are only the finest level elements along the blade traveling trajectory. This procedure will create new elements, and the number of newly-increased elements is proportional to the surface area being cut. And then we perform element cloning to represent the physical structure changes.

As shown in Figure 5(b), we first divide the element vertices into two sets. The vertices in each set are all on the same side of the cutting blade. And then, we clone

(a) Linear strain (b) Linear co-rational strain

Figure 7. Linear strain VS co-rational strain.

the two sets to form ghost vertex sets. The ghost vertices not only inherit the Doffs from the original sets but also have newly-redistributed Doffs. Therefore, we can obtain the new elements set by combining the original and ghost sets together.

Considering a cross-domain cutting situation, the element that have both FEM and EFGM properties is cut and cloned, the material-specific FEM property can be inherited as an invariant, while the EFGM property will be recomputed because of the update of material distance field arising from changing topological structure. And the updating operation is similar with the initialization operation.

5. CUDA-BASED IMPLEMENTATION

In implementation, we need to re-assemble the system equations in each simulation cycle because of modal warping and cutting operation (Algorithm 1). In order to accelerate system matrix assemble, we propose a matrix-free sparse data structure and design a CUDA-based Preconditioned Conjugate Gradient method on that basis.

Algorithm 1: CUDA based simulation.

Input: *velocity* $\mathbf{v_i}$ *and acceleration* $\mathbf{a_i}$ *of local domains at timestep* $\mathbf{i}$

Output: *displacement* $\mathbf{u_i}$ *for each local domain* D_i

1. Domain-parallel solving
for each *local domain* D_i,
1: update $\mathbf{f_i}$ with acceleration $\mathbf{a_i}$,
2: apply boundary condition to local system $\mathbf{S_i}$,
3: solve $\mathbf{u_i}$,
end for

2. Material-aware domain coupling
for each *overlap domain* O_i
5: assemble internal displacement $\mathbf{u_i^e}$,
6: solve corrective displacement $\mathbf{u_i}$,
end for

3. Compute corrective acceleration and force

4. Solve final solution
for each *local domain* D_i
7: update $\mathbf{f_i}$ with corrective force,
8: apply boundary condition to local system,
9: solve final displacement $\mathbf{u_i}$,
10: update acceleration $\mathbf{a_{i+1}}$ and velocity $\mathbf{v_{i+1}}$
end for

5.1. Matrix-Free Sparse Data Structure

In our method, system matrix is just a logical symbol, which is not physically stored on GPU memory and need not be assembled during the simulation loop. We employ an element-centric store structure (Figure 8). In this structure, each element has its own stiffness matrix, mass matrix, and Doffs, which is executed and updated in parallel to GPU independently. Figure 8 shows three elements of GPU memory layout. Benefiting from the structural regularity and uniformity of hexahedral element, each element only stores the common data index such as the local stiffness matrix ID.

5.2. System Assembling and Solving

To facilitate CUDA-based implementation, we propose an efficient sparse matrix-free structure-based storage strategy for the time-consuming assembling and iterative solving, which extends the standard Compressed Sparse Row mode and stores multiple nodes in the same column (Figure 9). Because each global coefficient in Lagrangian motion equations is the linear accumulation of the local coefficients, benefitting from the storage strategy, we can accumulate them using Equation (10) during solving process while not in the assembling procedure:

$$s_{(p,k)} = \sum_{C \in F, \tilde{k} \leftarrow k} \mathrm{SM}\,[\mathrm{C.sId}]\left[\mathrm{C.dofs}\left[\tilde{k}\right]\right] \qquad (9)$$

where $s_{(p,k)}$ is the value of our global logical matrix at (p, k), cell set F includes the cells that share the current specific degree of free, *SM* is a local matrix List containing the Local Stiffness Matrix in Figure 9, and $\tilde{k} \leftarrow k$ represents the transformation from the global Doffs to the local matrix index.

$$s_{(p,k)} = \sum_{C \in F, \tilde{k} \leftarrow k} \mathrm{SM}\,[\mathrm{C.sId}]\left[\mathrm{C.dofs}\left[\tilde{k}\right]\right] \qquad (10)$$

where $s_{(p,k)}$ is the value of our global logical matrix at (p, k), cell set F includes the cells that share the current specific degree of free, *SM* is a local matrix List containing the Local Stiffness Matrix in Figure 9, and $\tilde{k} \leftarrow k$ represents the transformation from the global Doffs to the local matrix index.

Then, we employ Preconditioned Conjugate Gradient method to solve the final linear system $y = Ax$, where A is the coefficient matrix derived from our simulation framework. Furthermore, we break up $y = Ax$ into separate equations. In our strategy, each row of A represents an independent degree of freedom (DoF). Therefore, we assign one CUDA thread to each DoF to conduct the parallel solving of $y = Ax$, wherein the boundary conditions can also be flexibly added to the system.

As for the changes of the linear system $y = Ax$ due to cutting, we can easily handle this by just cloning the physical quantities of the relevant Doffs, because the cutting operation only gives rise to the increase of Doffs.

DOI: 10.1002/cav

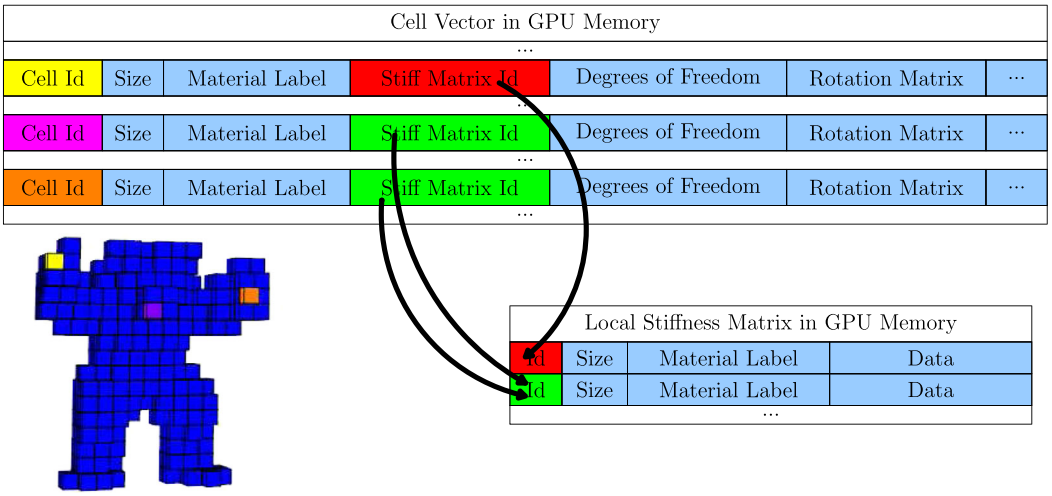


Figure 8. CUDA-based data structures.

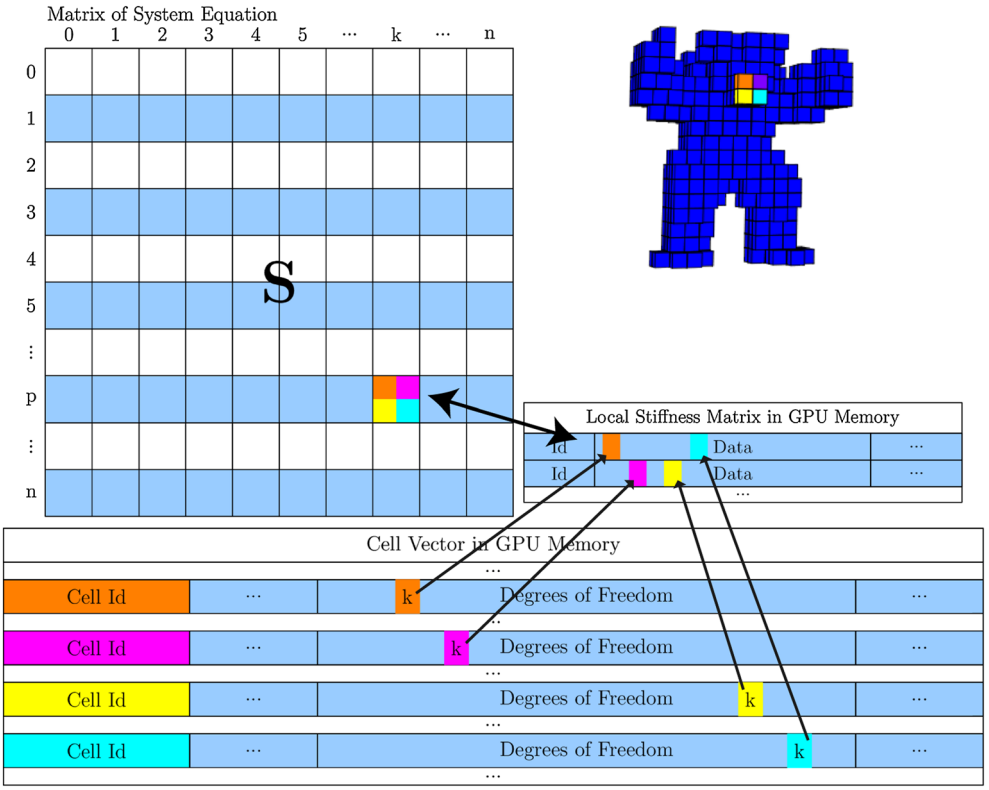


Figure 9. Dynamic assembling procedure.

6. EXPERIMENTAL RESULTS

We have implemented a prototype system using C++ and CUDA. All the experiments are run on a desktop PC with NVIDIA GeForce GTX 465 GPU, Intel Core i5 2.67 GHz CPU, and 2G RAM.

To validate the accuracy, efficiency, and applicability of our approach, we conduct experiments over different models. In order to show the coupling process of our approach, Figure 10 shows a pipeline of coupling two beams that have heterogenous materials. Figure 10(a) to Figure 10(d), respectively, show the material-specific FEM domains, the overlapping meshless domain, the results of step.1 in Algorithm 1, and the coupling results of the displacement fields in overlapping domain. Figure 10(e) shows the final simulation result, and

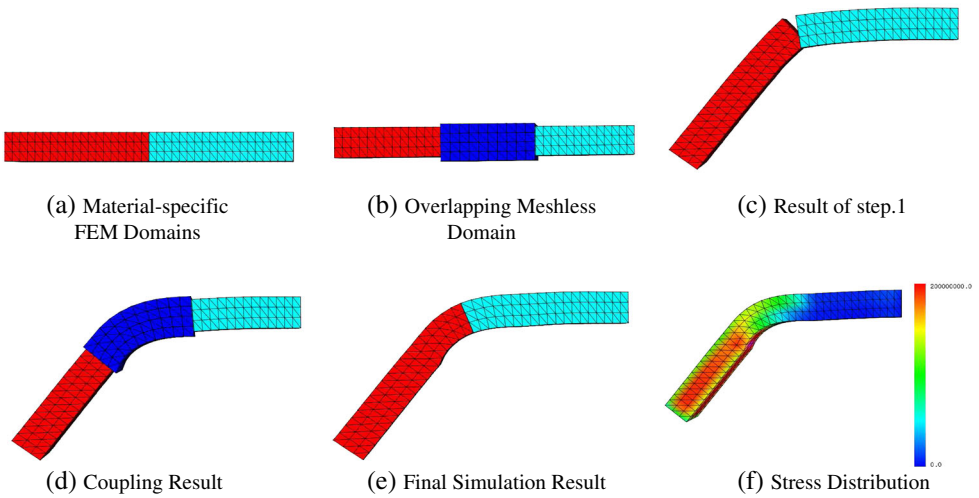


Figure 10. Couple simulation with two beams.

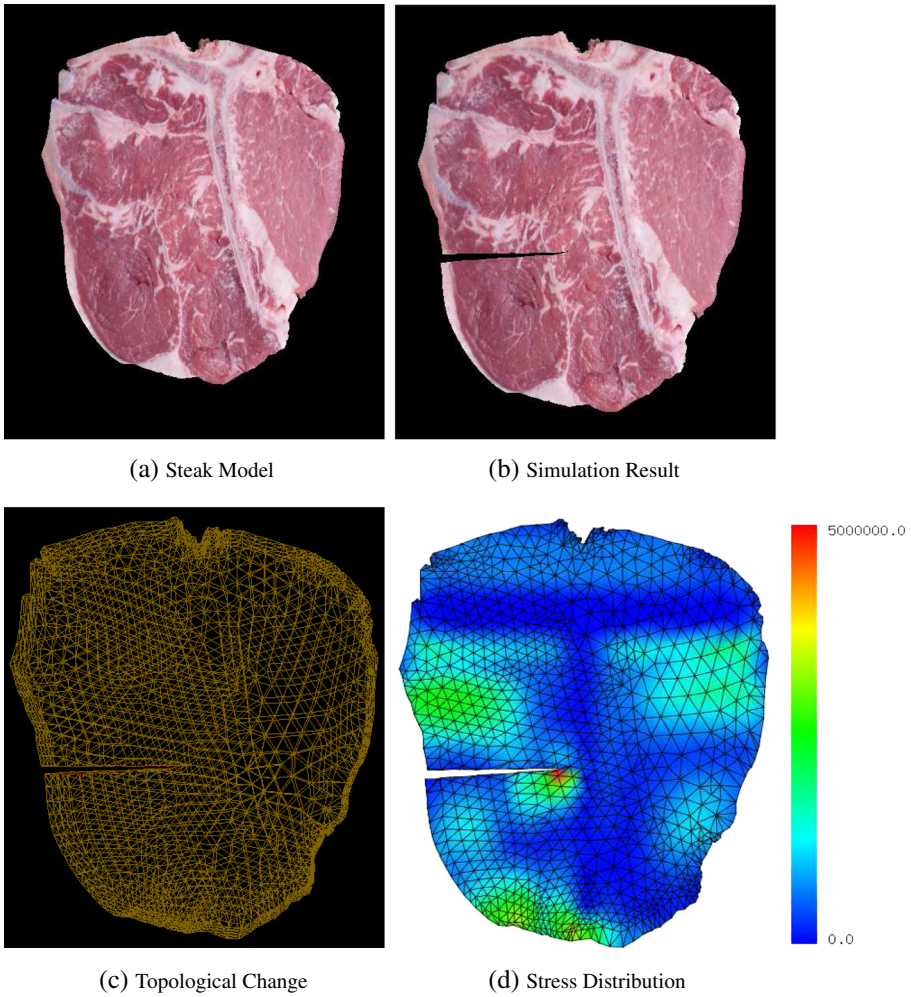
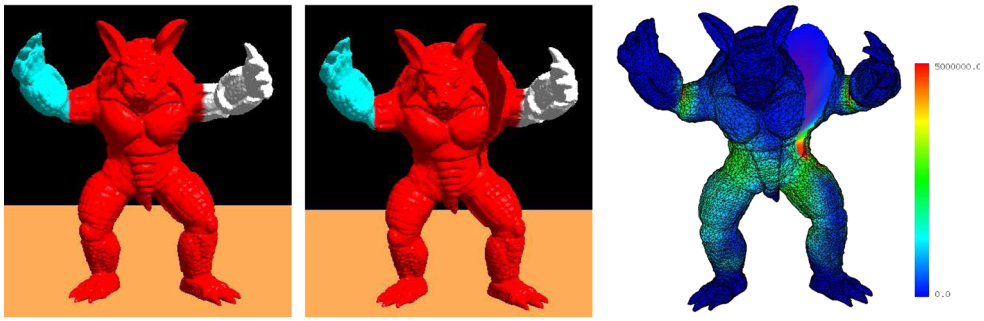


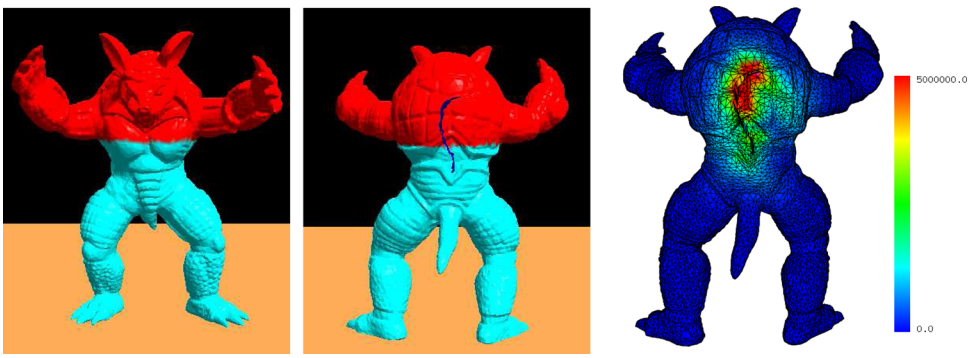
Figure 11. Cutting simulation over heterogeneous steak.

Comp. Anim. Virtual Worlds 2014; **25**:421–433 © 2014 John Wiley & Sons, Ltd.
DOI: 10.1002/cav



(a) Material Distribution (b) Cutting Operation in Single Domain (c) Stress Distribution

Figure 12. Single-domain cutting simulation over heterogeneous armadillo.



(a) Material Distribution (b) Cross-domain Cutting Operation (c) Stress Distribution

Figure 13. Cross-domains Cutting simulation over heterogeneous armadillo.

Figure 10(f) demonstrates the Von-Mises stress distribution corresponding to the final result. In this experiment, we respectively set the Young modulus of two FEM domains with 800,000,000 and 600,000,000, and the poisson ratios of the two domains are both set to be 0.33. Experiment indicates that our method is real-time, easy to operate, and has level-of-detail physical accuracy.

In order to verify the ability of our approach in handling the heterogenous object cutting, Figure 11(a) shows a steak model, which comprised several materials. Without loss of generality, we model the bone and the muscle as two material-specific FEM domains, wherein the (Young modulus,poisson ratio) are, respectively, set to be (900,000,000, 0.333) and (30,000,000, 0.333). Besides, we set those of the fat and soft tissue around the bone as (30,000,000, 0.333). Figure 11(b) shows the simulation result, Figure 11(c) shows the corresponding topological changes of the mesh arising from cutting operation, and Figure 11(d) demonstrates the Von-Mises stress distribution corresponding to Figure 11(b). It can be observed that our results can well conform with the realistic case.

Besides, to verify the generality of our approach, we conduct cutting simulation over the Stanford Armadillo model. Figure 12(a) illustrates the distribution of three materials in model. Figure 12(b) shows the result of a large-scale cutting operation in a single FEM domain. Figure 12(c) shows the stress distribution corresponding to Figure 12(b). Meanwhile, Figure 13(b) shows a cross-domain cutting operation, wherein an arc-shaped cutting path crosses two FEM domains and the overlap couple domain. Figure 13(a and c), respectively, shows the initial material distribution and the accompanying stress distribution. It demonstrates the universality and stableness of our method.

7. CONCLUSION AND DISCUSSION

We have detailed a CUDA-accelerated hybrid method to address a suite of research challenges in real-time and realistic deformation and arbitrary cutting simulation of heterogeneous objects. The originality of our approach is to model the physical system with the material-aware local FEMs and model the domain-interlinking with meshless EFT. Because each homogeneous sub-domain is relatively independent, local topological changes and domain-level

parallel numerical calculation can be efficiently accommodated. Our immediate efforts are geared towards incorporating blooding and suturing simulation into our current prototype system.

ACKNOWLEDGEMENTS

This research is supported in part by the National Natural Science Foundation of China (No. 61190120, 61190121, 61190125, and 61300067) and National Science Foundation of USA (No. IIS-0949467, IIS-1047715, and IIS-1049448). We would like to thank the anonymous reviewers for their very constructive comments and suggestions that help greatly improve this paper's quality.

REFERENCES

1. Wu X, Tendick F. Multigrid integration for interactive deformable body simulation. *Medical Simulation* 2004: 92–104.
2. Georgii J, Westermann R. A multigrid framework for real-time simulation of deformable bodies. *Computers & Graphics* 2006; **30**(3): 408–415.
3. Dick C, Georgii J, Burgkart R, Westermann R. Computational steering for patient-specific implant planning in orthopedics. In *VCBM*, 2008; 83–92.
4. Dick C, Georgii J, Westermann R. A hexahedral multigrid approach for simulating cuts in deformable objects. *IEEE Transactions on Visualization and Computer Graphics* 2011; **17**(11): 1663–1675.
5. Dick C, Georgii J, Westermann R. A real-time multigrid finite hexahedra method for elasticity simulation using cuda. *Simulation Modelling Practice and Theory* 2011; **19**(2): 801–816.
6. Miller KS, Joldes GR, Lance D, Wittek A. Total lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation. *Communications in Numerical Methods in Engineering* 2007; **23**(2): 121–134.
7. Taylor ZA, Comas O, Cheng M, Passenger J, Hawkes DJ, Atkinson D, Ourselin S. On modelling of anisotropic viscoelasticity for soft tissue simulation: Numerical solution and gpu execution. *Medical Image Analysis* 2009; **13**(2): 234–244.
8. Jerabkova L, Bousquet G, Barbier S, Faure F, Allard J. Volumetric modeling and interactive cutting of deformable bodies. *Progress of Biophysics and Molecular Biology* 2010; **103**(2-3): 217–224.
9. Forest C, Delingette H, Ayache N. Cutting simulation of manifold volumetric meshes. In *ESAIM: Proceedings*, 2002, EDP Sciences; 34–41.
10. Wicke M, Botsch M, Gross M. A finite element method on convex polyhedra. *Computer Graphics Forum* 2007; **26**(3): 355–364.
11. Noe KO, Sorensen TS. Solid mesh registration for radiotherapy treatment planning. *Lecture Notes in Computer Science* 2010; **5958**: 59–70.
12. Chao I, Pinkall U, Sanan P, Schröder P. A simple geometric model for elastic deformations. *ACM Transactions on Graphics* 2010; **29**(4): 8:1–38:6.
13. Fierz B, Spillmann J, Aguinaga I, Harders M. Maintaining large time steps in explicit finite element simulations using shape matching. *IEEE Transactions on Visualization and Computer Graphics* 2012-05; **18**(5): 717–728.
14. Fierz B, Spillmann J, Harders M. Element-wise mixed implicit-explicit integration for stable dynamic simulation of deformable objects. In *Proceedings of ACM SIGGRAPH*, New York, NY, USA, 2011, ACM; 257–266.
15. Müller M, Keiser R, Nealen A, Pauly M, Gross M, Alexa M. Point based animation of elastic, plastic and melting objects. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2004, Eurographics Association; 141–151.
16. Faure F, Gilles B, Bousquet G, Pai DK. Sparse meshless models of complex deformable solids. *ACM Transactions on Graphics (TOG)* 2011; **30**(4): 73.
17. Pauly M, Keiser R, Adams B, Dutré P, Gross M, Guibas LJ. Meshless animation of fracturing solids. In *ACM Transactions on Graphics (TOG),* Vol. 30, 2005, ACM; 957–964.
18. Steinemann D, Otaduy MA, Gross M. Fast arbitrary splitting of deforming objects. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2006, Eurographics Association; 63–72.
19. Gilles B, Bousquet G, Faure F, Pai DK. Frame-based elastic models. *ACM Transactions on Graphics* 2011-04; **30**(2): 15:1–15:12.
20. Belytschko T, Organ D, Krongauz Y. A coupled finite element-element-free galerkin method. *Computational Mechanics* 1995; **17**(3): 186–195.
21. Rabczuk T, Xiao SP, Sauer M. Coupling of mesh-free methods with finite elements: basic concepts and test results. *Communications in Numerical Methods in Engineering* 2006; **22**(10): 1031–1065.
22. Kim T, James DL. Physics-based character skinning using multidomain subspace deformations. *IEEE Transactions on Visualization and Computer Graphics* 2012; **18**(8): 1228–1240.
23. Barbič J, Zhao Y. Real-time large-deformation substructuring. In *ACM Transactions on Graphics (TOG),* Vol. 30, 2011, ACM; 91.

24. Yang Y, Xu W, Guo X, Zhou K, Guo B. Boundary-aware multidomain subspace deformation. *IEEE Transactions on Visualization and Computer Graphics* 2013; **19**(10): 1633.
25. Bosman J, Duriez C, Cotin S, *et al.* Connective tissues simulation on gpu. In *10th Workshop on Virtual Reality Interaction and Physical Simulation*, 2013, The Eurographics Association; 41–50.
26. Dionne O, de Lasa M. Geodesic voxel binding for production character meshes. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2013, ACM; 173–180.
27. Dick C, Georgii J, Westermann R. A hexahedral multigrid approach for simulating cuts in deformable objects. *IEEE Transactions on Visualization and Computer Graphics* 2011; **17**(11): 1663–1675.
28. Müller M, Gross M. Interactive virtual materials. In *Proceedings of Graphics Interface 2004*, 2004, Canadian Human-Computer Communications Society; 239–246.

AUTHORS' BIOGRAPHIES

Chen Yang received the BE degree in computer science from Tianjin University of Science and Technology in 2007. He is currently working toward the PhD degree in the State Key Laboratory of Virtual Reality Technology and Systems at Beihang University. His research interests include physics-based modeling and simulation, image-based rendering, and image processing.

Shuai Li received the PhD degree in computer science from Beihang University. He is currently a faculty member at the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. His research interests include computer graphics, real-time and realistic rendering, physics-based modeling and simulation, and medical image processing.

Lili Wang received the PhD degree from the Beihang University, Beijing, China. She is an associate professor with the School of Computer Science and Engineering of Beihang University, and a researcher with the State Key Laboratory of Virtual Reality Technology and Systems. Her interests include real-time rendering, realistic rendering, global illumination, soft shadow, and texture synthesis.

Aimin Hao PhD, professor, born in 1968, is professor of Computer School, Beihang University and vice director of State Key Laboratory of Virtual Reality Technology and Systems. His research interests are virtual reality, database application and information system development.

Hong Qin received the BS and MS degrees in computer science from Peking University, China. He received the PhD degree in computer science from the University of Toronto (UofT) in 1995. He is a full professor of computer science in the Department of Computer Science at State University of New York at Stony Brook (Stony Brook University). During his years at the University of Toronto, he received UofT Open Doctoral Fellowship. He was also a recipient of NSF CAREER Award from the US National Science Foundation (NSF), Honda Initiation Award, and Alfred P. Sloan Research Fellow by the Sloan Foundation. Currently, he serves as an associate editor for The Visual Computer, Graphical Models, and Journal of Computer Science and Technology. His research interests include geometric and solid modeling, graphics, physics-based modeling and simulation, computer aided geometric design, human-computer interaction, visualization, and scientific computing. Detailed information about him can be found from his website http://www.cs.sunysb.edu/~qin. He is a senior member of the IEEE and the IEEE Computer Society.