Pure appl. geophys. 165 (2008) 671–691
0033–4553/08/030671–21
DOI 10.1007/s00024-008-0328-6

Pure and Applied Geophysics

An Efficient System for Creating Synthetic InSAR Images from Simulations

XIAORU YUAN,[1] YINGCHUN LIU,[2] DAVID A. YUEN,[3] BAOQUAN CHEN,[6] TOMAS PERGLER,[4] and YAOLIN SHI[5]

Abstract—In this work we visualize tsunami and earthquake simulation results with graphics hardware acceleration. The rapid improvement in the computational power of graphics hardware and its programmability has made general computation on Graphics Processing Units (GPUs) very compelling. We generate Synthetic InSAR images using GPUs. Interference phenomena have formed the underlying theory for Interferometric Synthetic Aperture Radar (InSAR) in unveiling dynamical Earth movements. In our approach light path differences are defined by the surface values to be visualized. These path differences then modulate the lighting intensity to generate the interference patterns. We can interactively visualize surface deformation patterns by leveraging the computational power of GPUs. Our visualization method is applied to simulations of rupture fault displacements during the tsunamogenic earthquake events, which are vital to understanding the subsequent wave propagation. We also integrate the visualization results into Google Earth virtual globe to provide the geological context of the visualized regions.

Key words: Earthquake, GPU, InSAR, interference, tsunami, visualization.

1. Introduction

Underwater seismic events can produce tsunamis. Tsunamis can originate thousands of miles from land (traveling for hours before striking the coastline), or near shore (taking only minutes before impact). A massive undersea earthquake in the Indian Ocean measuring approximately moment magnitude (M_w) 9.3, known by the scientific community as the Sumatra-Andaman earthquake, hit the waters northwest of Sumatra on December 26th, 2004 and left more than 300,000 people dead or missing (USAID, 2006). It occurred along 1300 km of the oceanic subduction zone located 100 km west of

[1] Key Laboratory of Machine Perception Minister of Education, and Department of Machine Intelligence, School of EECS, Peking University, Beijing 100871, China. Correspondence E-mail: xiaoru.yuan@gmail.com
[2] South China Sea Institute of Oceanology, Guangzhou 510301, China.
[3] Department of Geology and Geophysics and Minnesota Supercomputing Institute, University of Minnesota at Twin Cities, Minneapolis 55455, U.S.A.
[4] Department of Geophysics, Faculty of Mathematics and Physics, Charles University, Prague 180 00, Czech Republic.
[5] Department of Earth Sciences, Graduate University of Chinese Academy of Sciences, Beijing, China.
[6] University of Minnesota of Tuni cities, Minneapolis 55455, U.S.A.

Sumatra and the Nicobar and Andaman Islands in the eastern Indian Ocean (STEIN and OKAL, 2005). This undersea earthquake is one of the largest earthquakes ever recorded on a seismograph. It was reported to be the longest duration of faulting ever observed, lasting between 500 and 600 s. It was large enough that it caused the free oscillation of the entire planet (PARK *et al.*, 2005). It also triggered earthquakes in other locations, even as far away as Alaska (WEST *et al.*, 2005). Simulations of fault displacements during a tsunamogenic earthquake event are vital to understanding the subsequent wave propagation (BILHAM, 2005; SUBARYA *et al.*, 2006). For example, numerical model simulations, combined with tidal-gauge and satellite altimetry data, reveal that wave amplitudes, directionality, and global propagation patterns of the Sumatra tsunami were primarily determined by the orientation and intensity of the offshore seismic line source and subsequently by the trapping effect of mid-ocean ridge topographic waveguides (TITOV *et al.*, 2005). Visualization has been very useful in interpreting such simulation results, comparing them with data obtained from other measures (VAN PUYMBROECK *et al.*, 2000), and helping scientists gain insights into such events (KRIKKE, 2005). In this paper we show the visualization of numerically generated post-seismic deformation by means of the computational power of burgeoning power due to graphics processing hardware.

The remainder of the paper is organized as follows. We briefly discuss graphics hardware in section 2. Details of our proposed interference inspired visualization method are given in section 3. Visualization applications of simulation data and discussions are described in section 4, followed by implementation details and performance measures in section 5. Finally we present our conclusions in section 6.

2. *Programmable Graphics Processing Units (GPUs)*

We exploit the recent advances in graphics hardware to accelerate our visualization applications. In this section, we briefly introduce the background of the current state of graphics hardware.

A Graphics Processing Unit (GPU) is a single chip processor with integrated rendering engines which is capable of processing a vast amount of geometry primitives. Although originally designed as a dedicated graphics rendering device for real-time rasterization of geometric primitives, modern GPUs are very efficient at solving computational intensive problems. The emergence of programmable GPUs makes general purpose computation (GPGPU) (GPGPU, 2006) feasible for a large range of complex scientific computational problems.

GPUs have been the most complex chips manufactured. One Nvidia GeForce 7800 GTX GPU chip has as many as 302 million transistors on board. The programmable pixel shaders of current GPUs can achieve a much higher theoretic peak performance than that of a currently available CPU. The ATI Radeon X1900 XTX has a claimed performance of 360 GFLOPS (Giga Floating Point Operations Per Second) of processing power (ATI, 2006), while a 3.7 GHz Intel Pentium4 SSE CPU can only achieve a theoretical peak of

14.8 GFLOPS (BUCK, 2004). A GeForce 7900 GTX chip can achieve a sustained 51.2 GB per second of memory bandwidth with a fill rate of 15.6 billion pixels per second and a vertex processing rate of 1.4 billion per second. As shown in Figure 2, it has 24 pixel shader pipelines and 8 vertex shader pipelines (NVIDIA, 2006). In general, current GPUs can offer more than an order of magnitude higher memory bandwidth and achieve up to an order of magnitude more computation speed up than current CPUs. Furthermore, GPU performance has improved faster than Moore's Law over the last decade. The graphics hardware performance is roughly doubling every six months. It takes the great advantages of utilizing GPUs for making scientific computation even more attractive.

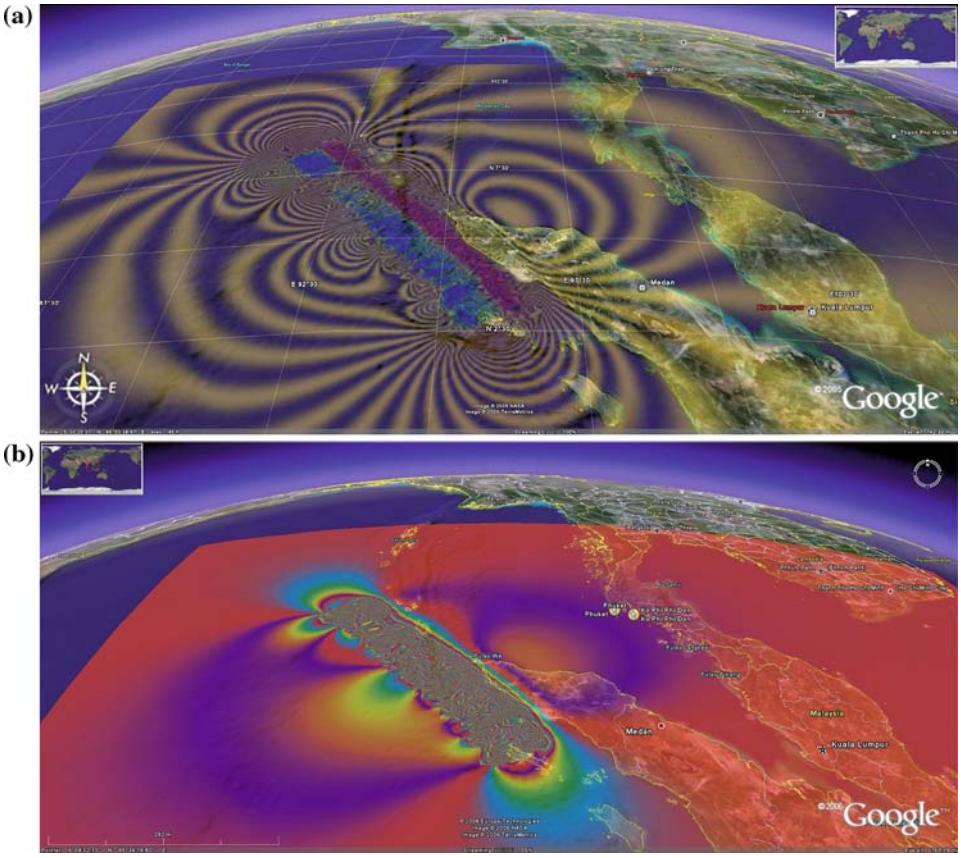

Figure 1
Our visualization of the vertical deformation of a fault rupture simulation of the Dec. 26th, 2004 Sumatra earthquake using GPUs with two different styles. (a) Red and blue areas are the rupture front and downdip slip regions with the most vertical displacement distances, respectively. Other regions have relatively smaller displacements. The interference pattern indicates the rate of change. Each interference period corresponds to 2.5 cm displacement in the vertical direction. (b) Using interference pattern. Each color cycle corresponds to 15.7 cm displacement in the vertical direction.

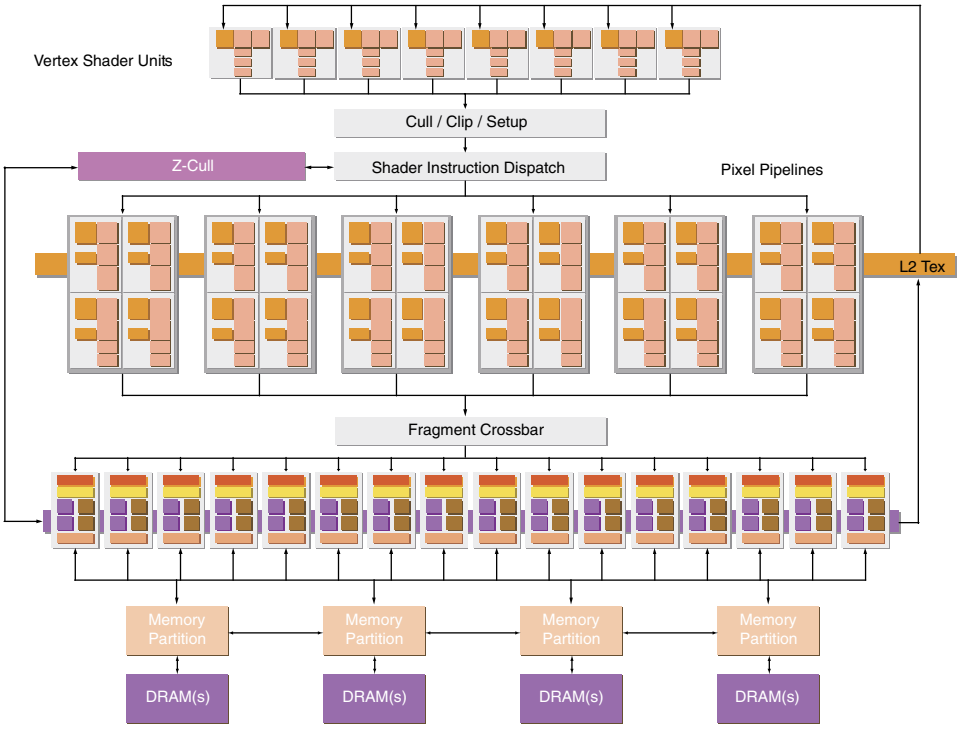

Figure 2
The chip layout of the Nvidia GeForce 7800 graphics processor. It has 24 pixel shader pipelines and 8 vertex shader pipelines.

GPUs are specialized computational devices. Computation in a GPU is done within the "Graphics Hardware Pipeline". The pipeline contains several steps in which the 3-D application sends a sequence of graphics primitives to the GPU to be processed by the programmable vertex processor and the programmable fragment processor subsequently. A representation of such a pipeline is shown in Figure 3.

To utilize GPUs to solve scientific computational problems, users need to transform the original computation setup to fit the graphics hardware pipeline, especially in the fragment processing stage. High level languages, (e.g., Cg, MARK *et al.*, 2002; GLSL, OpenGL, 2006), have been developed to facilitate implementing sophisticated computation on hardware. Cg is a high-level 3-D graphics programming language developed by Nvidia and Microsoft. Microsoft's implementation of Cg is High-Level Shading Language (HLSL), and both versions are identical. Cg is based on C++ and uses C++ syntax but is specialized for GPUs. OpenGL Shading Language (GLSL), also known as GLSlang, is another high level shading language based on the C programming language and independent of graphics hardware. In our implementation of this work, we have employed the Cg language.

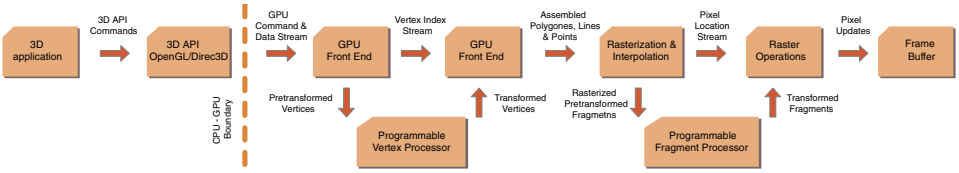

Figure 3
Graphics hardware pipeline.

Facilitated by the above high-level languages, programmable graphics processing units (GPUs) have been successfully applied to many domains beyond graphics, especially in scientific computation. We will demonstrate our applications of using GPUs to compute Synthetic InSAR images based on light interference for tsunami and earthquake simulation results.

3. Interference and INSAR

Interference phenomena are pervasive in nature. The physics of interference has been studied for more than 300 years. The first observation of interference was made by Robert Boyle in 1663 (GUENTHER, 1990), who observed what are now called Newton's Rings in which waves reflecting from two surfaces can interfere constructively and destructively. That interference pattern is associated with Newton because he performed a number of experiments on the effect. Young first generalized the principle of interference (YOUNG, 1802) and later developed his celebrated two-slit experiment (YOUNG, 1807). There are also rendering developments focused on simulating interference effects (DIAS, 1991; GONDEK, 1994; SMITS and MEYER, 1994; SUN, 1999). Our work is not intended to simulate the interference phenomena, but to use these patterns to visualize values and obtain results similar to InSAR images.

Interference is the interaction of two or more waves passing the same point. Constructive interference (Fig. 4(a)) occurs when the waves add in phase, producing a larger peak than either wave alone, whereas destructive interference (Fig. 4(b)) occurs when waves add out of phase, producing smaller peaks than one of the waves alone.

Consider the interference of two coherent monochromatic waves with wave functions ψ_1 and ψ_2,

$$\psi_n = \sqrt{I_n} e^{i(\mathbf{k}\mathbf{r}_n - \omega t + \varepsilon_n)}, \tag{1}$$

where I is irradiance, $\mathbf{k}$ is the propagation vector, $\mathbf{r}$ is the position vector, ω is the angular temporal frequency and ε is the initial phase. $n = 1, 2$. If they interfere coherently, then

$$\psi = \psi_1 + \psi_2 = \sqrt{I_1} e^{i(\mathbf{k}\mathbf{r}_1 - \omega t + \varepsilon_1)} + \sqrt{I_2} e^{i(\mathbf{k}\mathbf{r}_2 - \omega t + \varepsilon_2)}. \tag{2}$$

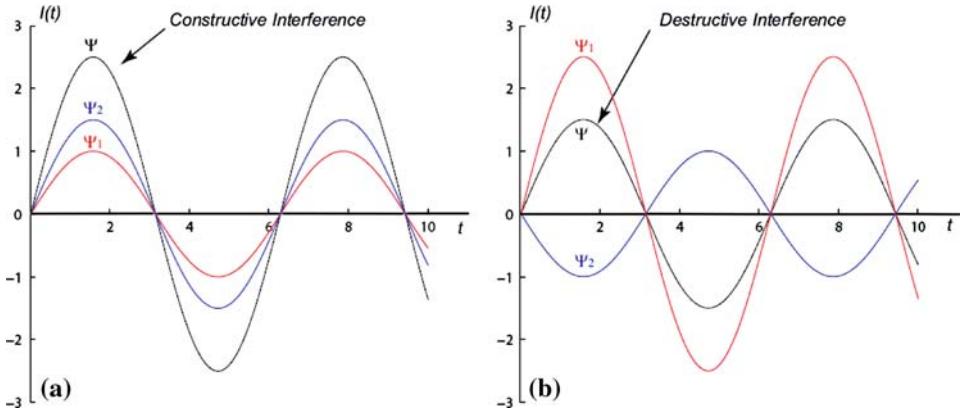

Figure 4
Light Interference. Red and blue curves are two coherent waves and the black curve is the interference results. (a) Constructive interference, (b) destructive interference.

The intensity is given by

$$I = |\psi|^2 = \psi\bar{\psi} = I_1 + I_2 + \sqrt{I_1 I_2}\cos\delta, \tag{3}$$

where

$$\delta = \mathbf{k}(\mathbf{r}_1 - \mathbf{r}_2) + (\varepsilon_1 - \varepsilon_2) = \mathbf{k}\Delta\mathbf{r} + \Delta\varepsilon. \tag{4}$$

δ is the phase difference arising from path length difference if the initial phases are the same.
If $I_1 = I_2 = I_0$ then

$$I = 2I_0(1 + \cos\delta) = 2I_0\cos^2\delta/2. \tag{5}$$

Constructive interference occurs when the phase difference between the two waves is an even integer multiple of π. Destructive interference occurs when the phase difference between the two waves is an odd integer multiple of π.

We apply the above interference patterns to visualizations of deformation on surfaces. We develop our visualization method by imitating the way interference is used to measure short distance variations by Newton's Rings. In our visualizations, light path differences are defined by the surface values to be visualized. These path differences then modulate the lighting intensity to generate the interference patterns. Assume for one point, the movement or deformation is d. We consider two coherent light waves traveling at a distance proportional to **d**. We define $\delta = 2\pi v\mathbf{d}$, where v is the **frequency** parameter used to control the frequency of the interference pattern. Then the interference intensity I will vary according to $1 + \cos(2\pi v\mathbf{d})$. Since the modulation term is a function with a period of 2π, each pair of peaks in the interference pattern indicates a difference of $1/v$ in **d**. We will refer to the value $1/v$ as the **resolution** of the subsequent interference rendering. This property is very useful for visualization purposes.

There are a few variations we could apply to use such interference patterns for visualization. In Figure 5, we demonstrate such visualizations on an artificially created data set. The data set is a plane with two peaks generated with the peak function in Matlab. As shown in Figure 5(a), the surface patch is a square with a size 255×255. The vertical deformation of two peaks varies from -0.0655 to 0.0811. The vertical elevation in the figure is exaggerated by 1000 times to make it visible.

As illustrated in Figure 5(b), we modulate the deformation value to an interference pattern and display the modulated values as intensity values in gray levels. With a v value of 100, (**resolution 1/v = 0.01**) each neighboring band indicates a deformation difference of 0.01. To display the modulated values, we scale them to be in the range of [0,1] by using the formula of $((1 + \cos(2\pi v\mathbf{d})) / 2)$. The same scaling is always applied through out this paper.

In Figure 5(c), we apply different frequencies for each color channel to visualize the deformation. Three frequencies of (100, 110 and 120) are applied. In this way, the result closely resembles the interference patterns generated with white light. The color order of transition regions indicates the decrease or increase of the value to be visualized. We also apply a color map (Fig. 5(g)) on top of the interference pattern as shown in Figure 5(d) with a formula of $colormap(normalize(\mathbf{d}))\cdot((1 + \cos(2\pi v\mathbf{d})) / 2)$. The function $colormap(\,)$ maps

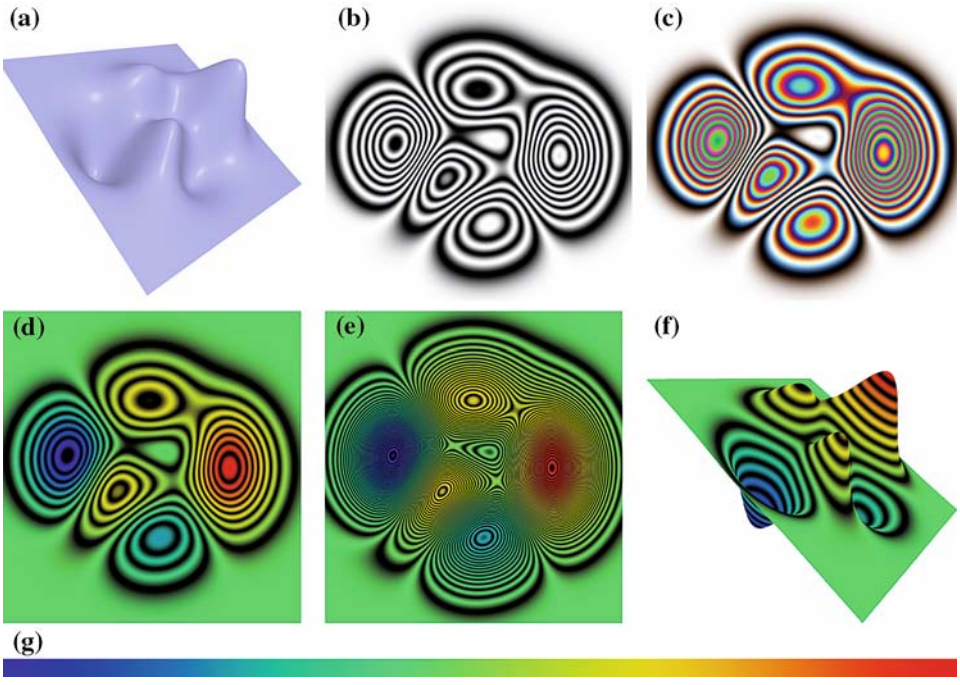

Figure 5
Interference patterns generated with variations. (a)–(d) all represent a surface with two peaks rendered in (e). In (f) the interference patterns are mapped to the 3-D surface.

values between 0 and 1 to a specified color map. The function *normalize*(**d**) linearly map **d** to range of [0,1]. Such an operation is also applied for Figure 1(b). Figure 5(e) has the same configuration as Figure 5(d) except that a higher frequency v value of 500 is applied. More details in low deformation regions can be revealed with higher frequency values. Such a visualization pattern is applied in Figure 1(a). In Figure 5(f), the rendered interference pattern is mapped on the exaggerated geometry model. We can also apply a color map to each cycle of interference pattern with formula of $colormap((1 + \cos(2\pi v\mathbf{d})) / 2)$. Since the function $(1 + \cos(2\pi v\mathbf{d})) / 2)$ is not a monotonic function, a color map will be mapped twice within one single interference cycle in two opposite directions. We can convert it into monotonic function using a formula of $colormap(((1 + \cos(\pi v\mathbf{d}))/2) \cdot sign((1 + \sin(\pi v\mathbf{d})) / 2))$. Function $sign(x)$ returns the value 1 for positive variables while it returns –1 for negative variables. With the factor $sign((1 + \sin(\pi v\mathbf{d})) / 2)$, the whole function has a period of π. In this case, each color period still indicates a difference of $1/v$ in **d**. This formula is used to visualize an earthquake simulation in Figure 1(b). In all the above renderings, the computation of interference modulation is performed in GPUs. We will discuss the details of the computation at the end of this paper after we present all visualization results.

The above interference-based method generates similar images to Interferometric Synthetic Aperture Radar (InSAR) (Fig. 6). Indeed, interference phenomena have formed the underlying theory for InSAR in unveiling earth movements. Radar satellites constantly shoot beams of radar waves towards the Earth and record their return after they have bounced back off the Earth's surface (MASSONNET, 1993). The phase shift of the bounced wave could be used to indicate the deformation that has occurred. Such techniques are utilized to provide direct and precise measurement of the vertical changes in ground level. The strength of InSAR lies in its ability to provide observations of ground displacements with a precision as high as a few millimeters with 20 meters of

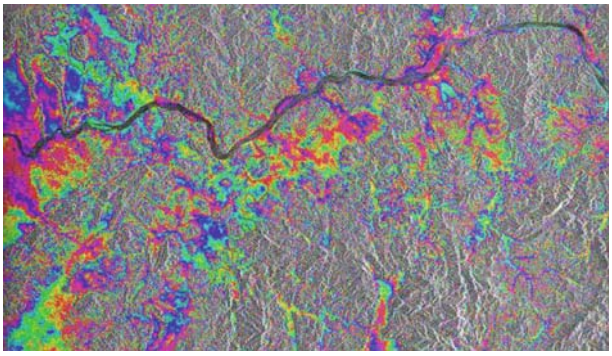

Figure 6
A sample interferometric image obtained with the Synthetic Aperture Radar (SAR) instrument on-board the ESA ERS satellites. Each cycle of the colors represents a change in the ground height which is dependent on the satellite geometry. The image is from the European Space Agency. (http://earth.esa.int/images/INSI/).

spatial resolution and cover hundreds of kilometers in spatial extent. InSAR has revolutionized the seismic and volcanic study.

Figure 6 is a sample interferometric image obtained with the Synthetic Aperture Radar (SAR) instrument on-board the ESA ERS satellites. The data are collected when the ERS-1 and ERS-2 satellites were flying in "tandem-mode" observing the same part of the ground with a 1-day interval. Each cycle of the colors (for example, going from yellow to purple to turquoise and back to yellow again) represents a change in the ground height, which is dependent on the satellite geometry. The introduction of InSAR makes the production of displacement maps of an entire region with sub-centimeter accuracy possible. The InSAR process involves the acquisition of two or more single images from an airplane or satellite. The radar sensors carried aboard remote-operates in the microwave range of the electromagnetic spectrum. To create an interferogram, two images must be acquired at different points in time.

After the powerful 2004 Sumatra earthquake, significant efforts have been devoted to measuring, simulating and visualizing the scenario and understanding this largest seismic event since the introduction of digital seismology (Ammon *et al*., 2005; Krikke, 2005; Lay *et al*., 2005). Steven Ward (2006), a geophysicist produced an animation of the Indian Ocean tsunami hours after the event. Ammon *et al*. (2005) studied the rupture process of the Sumatra-Andaman Earthquake. The research results showed that the earthquake was initiated slowly, with small slip and a slow rupture speed. Then the rupture expanded quickly toward the northwest, extending 1200 to 1300 kilometers along the Andaman trough. Lay *et al*. (2005) and Bilham (2005) also indicated a slow slip. The InSAR imaging is also perfect for measuring the Earth's deformation during a tsunamogenic earthquake event. There is a need to carry out a rapid comparison between modeling and observations from InSAR images. Responding to such demands, our research focuses on visualizing the simulated rupture process of the earthquake. A major motivation for developing this interference inspired visualization is its similarity to InSAR, which is a widely applied remote sensing technique for measuring deformation of the Earth's surface. The interference patterns generated in our approach are easier to understand for geophysicists who have previous experience with InSAR images. We generate such Synthetic InSAR images from the simulation results to allow researchers a quick assessment of the results. It is desirable for researchers to have quick access to InSAR images for comparison purposes. Since conversion of an InSAR image to an elevation map is still computationally expensive, our method provides a faster way for direct comparison.

4. Visualization for Tsunami Event Simulations

In this section we illustrate how our interference inspired method could be applied to various geophysical simulation data sets to better understand tsunami events. For the sake of completeness, we also briefly discuss the models used in these simulations.

4.1. Case 1: Visualization of a Hybrid Simulation Model

In this first numerical simulation, we use a hybrid method which is basically set up from finite-element methods in a 2-D area with a Fourier series decomposition as the third dimension. We deal with the elastic problem for simplicity.

$$\nabla \cdot \tau + \rho_0[(\nabla \cdot \boldsymbol{u})\boldsymbol{g}_0\boldsymbol{e}_z - \nabla(\boldsymbol{g}_0\boldsymbol{e}_z \cdot \boldsymbol{u})] = 0, \tag{6}$$

$$\tau - \lambda(\nabla \cdot \boldsymbol{u})\boldsymbol{I} - 2\mu\varepsilon(\boldsymbol{u}) = 0. \tag{7}$$

We can decompose the 3-D problem this way if the equation coefficients are independent in one horizontal direction and also if it is possible to rotate the domain according to the rupture. In this case the spectral decomposition is used as the y axis and the rupture function is placed to the computation domain along this axis. The rupture function has an important role in our solution technique, thus we explain this part in greater detail.

This function describes displacements on the 2-D fault itself. It has to be defined to be continuous on some part of the 3-D domain with non-zero measurement because of our mathematically weak problem formulation. It means this function is dependent on all three coordinates $f(x, y, z)$ and when we transform it to the spectral domain we have function $f(x, k_y, z)$ where k_y is now only a parameter. After solving simultaneous equations (6) to transform to the spectral domain we solve just 2-D problems with multiple parameters k_y. We just use backward transformation of 2-D FEM solutions to transfer the solution back to the 3-D domain. The numerical simulations were performed by, 2-D finite-element code together with the Fourier decomposition method to create adapted mesh. The GMRES method with ILU preconditioning is used as the linear solver. The code was written in C-programming language. To increase code effectiveness, OpenMP parallelization was used for the backward transformation and for the creation of surface mesh.

The domain of the subduction model is sized 1000 km $\times$ 1000 km $\times$ 600 km. The fault strike is 0°, the dip is 60° and the rake is $-90°$. The fault is 50 km deep and amplitude is 10 meters, its length in y coordinate is 100 km and its second dimension is 50 km. For the Fourier decomposition 600 k_y parameters were used. The computation mesh has 563 $\times$ 564 $\times$ 100 grid points. With respect to the concentration of points close to the rupture, it was worthwhile to increase the number of points to 1001×1001 during recalculation of equidistant rotated surface mesh. The computation was run at the Minnesota Supercomputing Institute on Altix workstations, where parallelism was utilized for the most time-consuming parts of the computations.

In Figure 7 we visualize the computational results using interference patterns. A color map (Fig. 7(d)) is mapped to each period of the interference pattern. Figures 7(a), (b) and (c) show the x, y and z components of the fault deformation simulations, respectively. In each group, the top row shows the interference-based visualization with v value of 2, 4, 8, 16 and 32, respectively. The bottom row images are zoom-ins of the center part of their corresponding top row images. Note that the boundary between the color blue and red is due to the two ends of the texture for which we used different colors. If a cyclic texture is

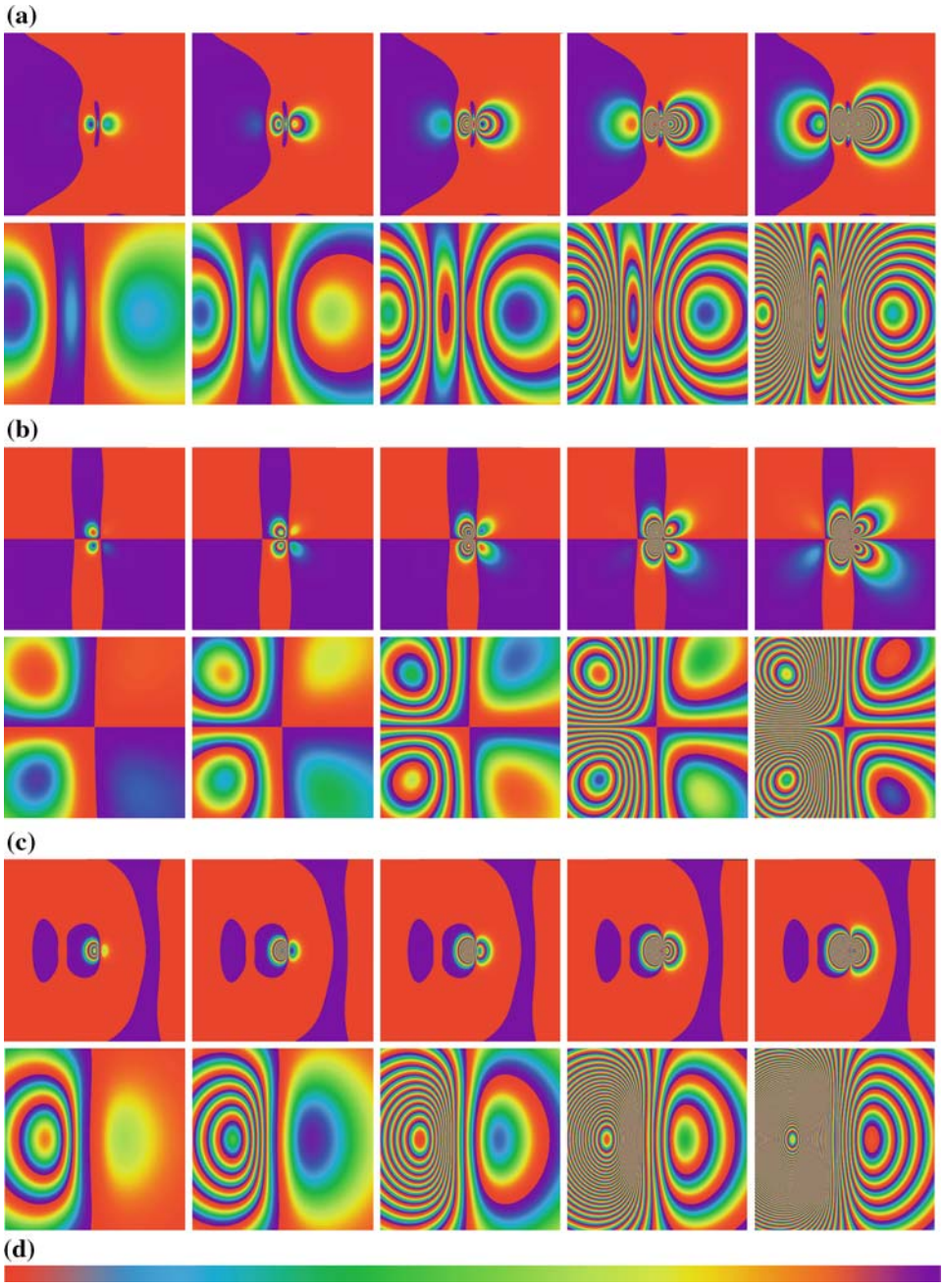

Figure 7
(a) x component, (b) y component, (c) z component of the fault deformation simulated by a Hybrid Simulation Model. In each group, the top row shows the interference-based visualization with v value of 2,4,8,16 and 32, respectively. The bottom row images are zoom-ins of the center part of their corresponding top row images.

used (in Fig. 9), there will be no such sharp color boundaries. Note in all examples shown in this paper, the displacement calculations are carried out in a Cartesian coordinate system. When the curvature of the Earth is considered, it is also possible to do the interference pattern computation in a spherical coordinate system, in which case, latitude and longitude are used to substitute vertical and horizontal values in the Cartesian geometry.

4.2. Case 2: Earth Rupture Simulations of Sumatra Earthquake

In the second case, we use the PSGRN/PSCMP code (OKADA, 1992; WANG *et al.*, 2006) to calculate coseismic and postseismic deformation for layered rheological media. In recent years continuous GPS data sets have been frequently used. With this data researchers can study the coseismic and postseismic deformation after the earthquake. In fact, the Earth can be described using two different material models in which the upper layer is elastic and the deeper is viscoelastic. The postseismic deformation is a very complicated process, especially as some factors are neglected in relaxations with long time duration.

The PSGRN/PSCMP code is based on the extension of the elastical dislocation theory with the gravity effect. The theory is applied to the governing equations for the infinitesimal static deformation in a self-gravitating, hydrostatically prestressed Earth. This approach extended the earlier elastic dislocation theory with the gravity effect reflected in the following equations:

$$\nabla \cdot \Gamma + \rho \nabla (\mathbf{u} \cdot \mathbf{g}) - \mathbf{g} \nabla \cdot (\rho \mathbf{u}) = 0, \tag{8}$$

$$\nabla^2 \psi - 4\pi G \nabla \cdot (\rho \mathbf{u}) = 0, \tag{9}$$

where Γ is the Lagrangian incremental stress tensor, $\mathbf{u}$ is the displacement vector, ψ is the Eulerian incremental potential, g is acceleration due to gravity, G is the gravitational constant, and ρ is the density. For an isotropic and elastic medium, Hook's linear constitutive relation between the stress and strain is valid,

$$\Gamma = (\lambda \nabla \mathbf{u})\mathbf{I} + \mu[\nabla \mathbf{u} + (\nabla \mathbf{u})^T], \tag{10}$$

where λ and μ are the two Lame constants, $\mathbf{I}$ is the unit tensor, and $(\nabla \mathbf{u})^T$ denotes the tensor transpose of $\nabla \mathbf{u}$. With consideration of various practical mathematical and geophysical issues, the governing equation can be reduced to:

$$\nabla \cdot \Gamma + \rho g \nabla u_z - (\rho \nabla \cdot \mathbf{u} + \frac{\rho^2 g}{\kappa} u_z) g e_z = 0, \tag{11}$$

where κ is constant and $\mathbf{g} = g e_z$ has been used. This approach considers the coupling between the deformation and the Earth's gravity field and produces the postseismic deformation data. The output includes the complete deformation field consisting of 3 displacement components, 6 stress (strain) components and 2 tilt components, and the

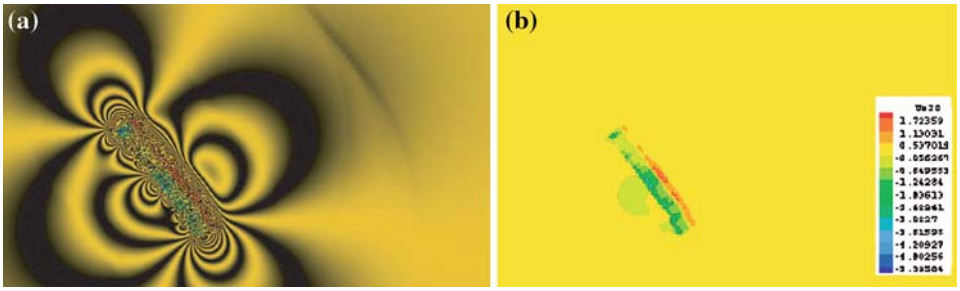

Figure 8
Comparing Interference Visualization with linear color mapping. (a): Interference patterns modulate the color map. (b): Color mapping using Tecplot. The field is a simulation of the displacements of the fault of the Sumatra earthquake on Dec. 24, 2004.

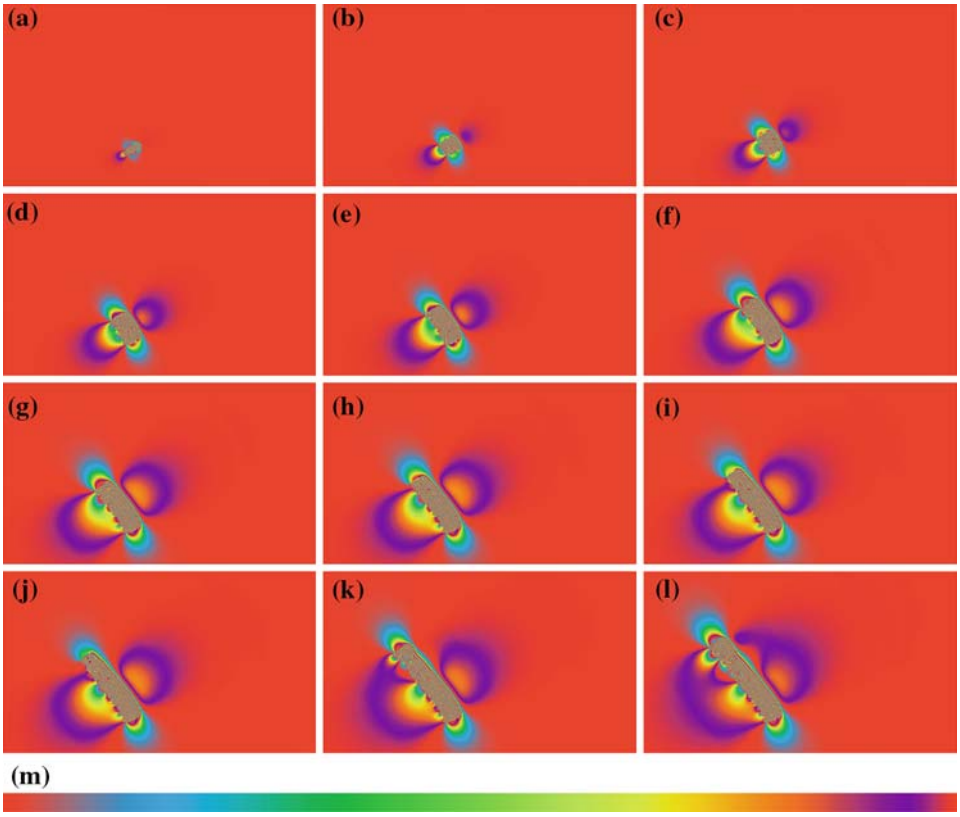

Figure 9
Interference Pattern Visualization of z displacement component of a simulation of a coseismic fault of a region of 0–14°N, 86–110°E. (a)–(l) 12 time steps are visualized. (m) The color map used in this figure.

geoid and gravity changes. For details of this computational model and implementation, please refer to Wang's recent work (WANG *et al.*, 2006). In this visualization work, we are mainly interested in visualizing the displacement components generated by such simulations.

The study field is a much larger region simulation for the Sumatra earthquake of Dec. 26, 2004. The fault length of the earthquake was 1300 km. The magnitude was M_w 9.3. The epicenter coordinates were 3.1°N;95.7°E. The fault strike was 310° and the dip was 10°. The fault width was 100 km. The displacements of the fault were computed by coseismic dip. The program code PRCMP3.2 and PSGRN3.2 was written by RONG-JIANG WANG (2006). We set the material of the fault to be elastic. The size of the field is 0–14°N, 86° –110°E. The rupture of the fault is 23 time steps along the strike and 7 layers in the vertical direction. The grid is 121×121 evenly spaced points. The maximal displacement is 13 m.

We visualize the vertical displacement component of such time-varying deformation surfaces with the parameter of $v = 50$, equivalent to 0.02 m in resolution. In Figure 8 we compare the visualization result using traditional color mapping and the interference pattern. Our method can clearly illustrate the details of deformation computed. In Figure 8(b), a color map is used to map the z displacement of the fault in the Sumatra earthquake simulations using TecPlot. In Figure 8(a), we modulate the visualization color mapped image with our interference pattern. The rings indicate the rate of the displacement change. The red color regions have positive vertical displacements, and the green color regions have negative vertical displacements. It is obvious that traditional color mapping methods problematically deal with such data sets due to a lack of differentiable color levels. The interference pattern clearly visualizes the deformation propagation in the surrounding areas.

In Figures 9(a)–(l), we illustrate 12 different time steps of the fault simulation visualization. The vertical displacements are visualized. The color mapping used in this visualization is shown in Figure 9(m). The same visualization is also mapped to Google Earth to show the geological context that is shown in Figure 1(a). From the visualization, it is clear that the rupture developed from the epicenter and expanded quickly toward the northwest which is the *a priori* constraint we set on the modeling (AMMON, 2005; BILHAM, 2005; LAY *et al.*, 2005). The fault can be categorized as a reverse fault. Data collected at 60 Global Positioning System (GPS) sites in Southeast Asia show the crust deformation caused by the 26 December, 2004 Sumatra-Andaman earthquake at an unprecedented large scale (VIGNY *et al.*, 2005). Small but significant coseismic jumps are clearly detected more than 3,000 km from the earthquake epicenter. Such small deformations are very obvious in our visualization. The nearest sites, still more than 400 km away, show displacements of 10 cm or more. Here we show how the rupture plane for this earthquake must have been atleast 1,000 km long and that nonhomogeneous slip is required to fit the large displacement gradients revealed by the GPS measurements. Our kinematic analysis of the GPS recordings indicates that the centroid of the released deformation is located at least 200 km north of the seismological epicenter. It also provides evidence that the rupture propagated northward with sufficient speed such that stations in northern

Thailand reached their final positions less than 10 min after the earthquake, ruling out the hypothesis of a silent slow seismic rupture.

4.3. Case 3: Tsunami Wave Simulation of South China Sea

We have also applied our visualization method to the computation of wave height generated by the tsunamogenic earthquake in the South China Sea (LIU *et al*., 2007). The trenches of the South China Sea stretch in a north to south direction, from south of the Taiwan Island, passing west of Luzon and extending to the west of Mindanao. The sea-bottom topography of the Manila trench is very complicated. The large variation of water depth and sea-bottom topography indicates the possibility of an earthquake-tsunami in that region. The Manila trench is the fault-graben trough dominated by normal fault, on the boundary of the Epicontinental graben system of the South China Sea and the Philippine Island-arc fault-fold system. The western part of the Manila trench fault zone is formed by the steepened fault and depression of the Central oceanic basin of the South China Sea. In terms of the fault movement, the Manila trench fault zone is the region most likely to have experience a tsunami in the South China Sea.

Based on such reasoning, a simulation of the sea-water wave height in that region has been conducted. As shown in Figure 10(a), we apply interference based visualization to the wave-height data computed. The wave height is stored in its absolute value. A color texture is mapped to each period of the interference pattern. The regions with denser interference fringes are associated with higher wave height and in larger danger of a tsunami. Figure 10(b) is a color texture mapped to the whole data range and then modulated by the interference pattern. Figure 10(c) uses a direct color mapping without interference pattern enhancement.

4.4. Integrating Interference Visualization Results with Google Earth

The tsunami and earthquake data sets visually at our disposal are associated with specific regions of the globe. It is therefore very important that we have the contextual geological information necessary for detailed visualizations. Therefore, we integrate our visualization results into GOOGLE EARTH (2007) using the image overlay function provided by its software. Google Earth is a server-client based virtual globe program (http://earth.google.com/). It maps the entire earth by pasting images obtained from satellite imagery, aerial photography and GIS over a 3-D globe. Google Earth allows users to search for addresses, enter coordinates, or simply use the mouse to browse to a location. Google Earth also has digital terrain model data collected by NASA's Shuttle Radar Topography Mission. Users can directly view the geological features in three-dimensional perspective projections, instead of as 2-D maps.

Many scientists have found Google Earth to be a powerful tool for scientific visualization (BUTLER, 2006; NOURBAKHSH, 2006). In our experiment, we use the image overlay function of the Google Earth software to integrate our visualization results into a

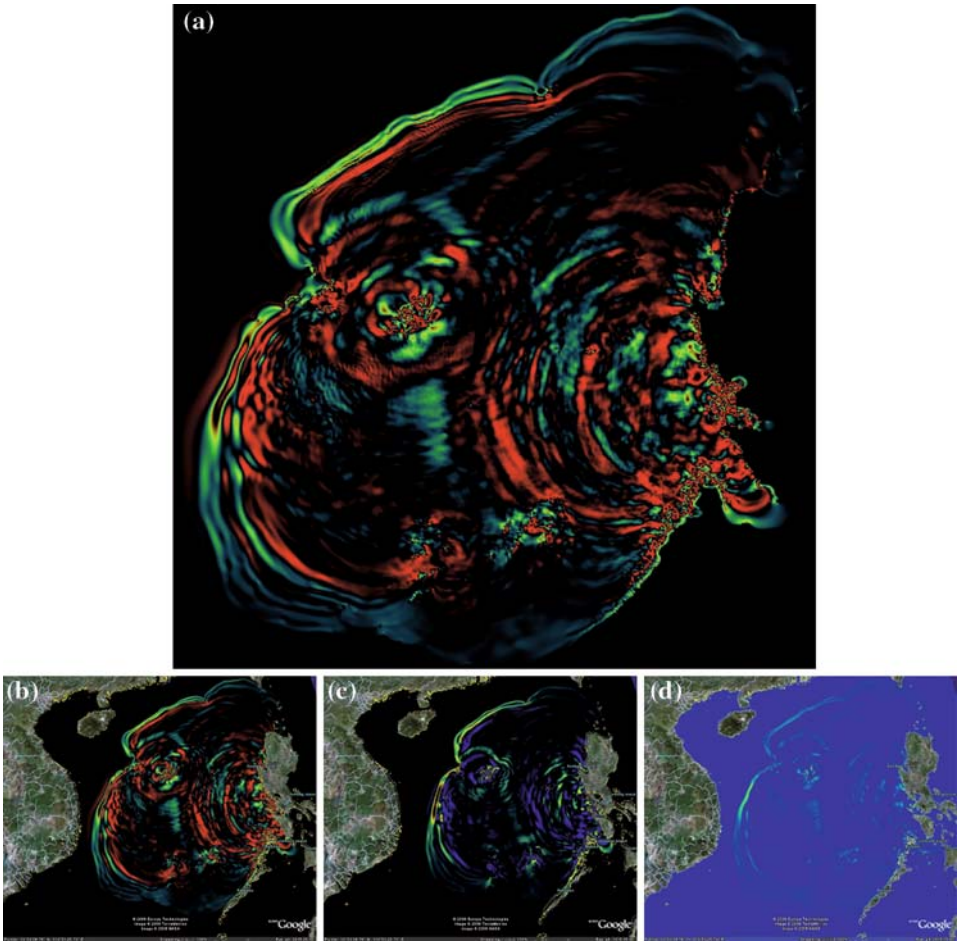

Figure 10
Visualization of wave height generated by a tsunamogenic earthquake in the South China Sea. (a) and (b) color mapped to each period of the interference pattern, (c) color mapped to the whole data range and then modulated by interference pattern, (d) direct color map.

virtual globe context. Figure 1 illustrates the integration of our visualization of the fault simulation of the Sumatra earthquake with Google Earth. The interference pattern visualization images are overlayed in Google Earth. The geological context of the terrain is rendered in 3-D and local cities are annotated. Such visualizations help users to associate the scientific data with a geological context which is familiar to ordinary users. In Figure 10, 3 other examples are also shown.

Figure 11 shows how users can navigate the virtual globe with integrated visualizations. Note that in the most zoomed-out image, Google Earth provides the recorded earth quake information indicated by a red dot in the studied region. Such

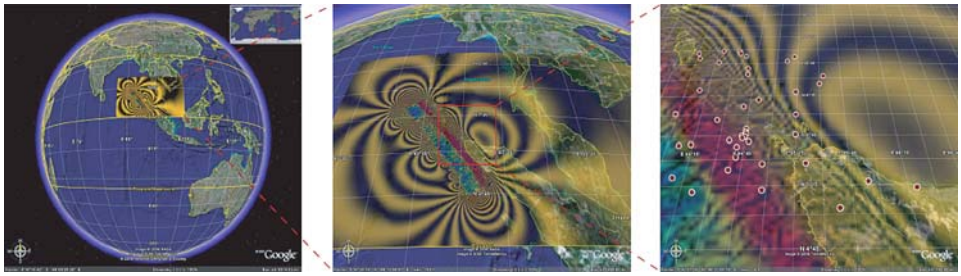

Figure 11
Navigation of Google Earth with integrated visualization results.

information is valuable for researchers seeking to understand the event. The software provides a convenient way for researchers to share their results. The visualization results we embed in Google Earth could be in a data exchange format (KML) together with the geological context information. Such meta data could be directly opened by other parties who also have the software. The Google Community service could make these visualization results available to the public, allowing the sharing of research results between research groups.

5. *Implementation Details and Performance*

All visualization computations including rendering have been performed on a Dell Precision 530 workstation with single Intel Xeon 2.20 G Hz CPU, 1 GB RAM, 4 × AGP motherboard and a 256 MB GeForce 6800 Ultra graphics card. Cg is used for GPU programming. We implement our rendering pipeline with deferred shading to improve interactivity for large data set visualization. The concept of deferred shading was introduced by DEERING *et al.* (1988) and was later used by PixelFlow (MOLNAR *et al.*, 1992.) With advances in commodity graphics hardware, deferred shading is becoming practical for normal rendering (NVIDIA, 2004; SHISHKOVTSOV, 2005).

For the geophysical simulation data we are dealing with, there are multiple propensities associated with each grid point. Before the deferred shading, we render all values into multiple textures illustrated as the first rendering step in Figure 12. Current Multiple Render Targets (MRT) features enable us to simultaneously render up to four textures in a single pass. With one color channel in a texture reserved for the valid rendered area, there are 15 channels available for information storage. We use floating point textures to store the rendered results. After the first rendering pass, we create intermediate textures with color channels storing deformation values. Users can freely select the surface properties they wish to visualize in the second deferred shading pass illustrated as the second rendering step in Figure 12. The program computes the

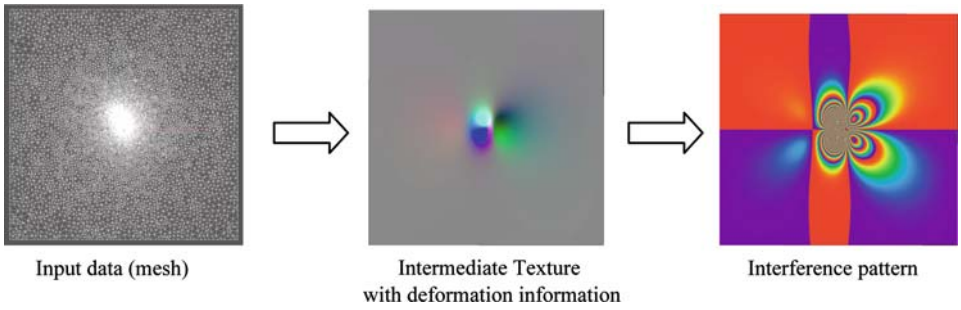

Figure 12
Rendering pipeline for interference based-visualization.

Table 1

Performance of our rendering method for various models. Two different screen sizes are used to test the performance. T_{geo} is the run time before the deferred shading. T_{def} is the time for the deferred shading.

Model	Grid Size	Screen Size	T_{geo}(ms)	T_{def} (ms)
Peaks (Fig. 5)	256 × 256	512 × 512	22	5.5
	256 × 256	1200 × 900	23	8
Sumatra earthquake (Fig. 11)	121 × 121	512 × 512	11	5.6
	121 × 121	1200 × 900	12	7

interference patterns using a fragment shader together with an optional color map lookup. We leverage the current commodity graphics hardware to perform the computation and rendering. Our method is very efficient. Table 1 shows the timing for the rendering of different data sets. Two different screen sizes are used to test the performance. T_{geo} (in milliseconds) is the run time before the deferred shading. T_{def} (in milliseconds) is the time for deferred shading. Note that because the deferred shading is executed on the final frame image, it is independent of model resolution and dependent only on the screen resolution as shown in the table.

6. Conclusions and Future Endeavors

We present a novel visualization method for visualizing dynamically changing surfaces, which is one type of highly dynamic range data (YUAN *et al.*, 2005, 2006, 2007). Our method is superior to traditional color mapping and can illustrate subtle but important surface value variations. We can interactively visualize surface deformation patterns by leveraging commodity graphics hardware to achieve interactivity. Our method is simple, easy to implement and very effective at illustrating subtle but critical surface changes to surface features in deformation processes. As compared with the

visualization using existing software tools, we gain advantage in the quality and speed with graphics hardware acceleration. The integration of our seismic visualization with Google Earth provides an easy way to include global geological context. Users can easily exchange research results using this tool.

It is possible to combine interference patterns with other existing visualization methods. In the future we will test our method on more complex models which incorporate bends and other geometric complexity. We also plan to extend our work to other simulation data sets. For one immediate step, we plan to use the same setting of the InSAR images and generate interference patterned images for direct comparison with InSAR satellite images. We would also like to compare our calculated displacement values with the observational values.

Acknowledgement

The authors would sincerely thank Google Inc. for providing the software Google Earth Pro. to Peking University through the Google Earth Education Initiative. We wish to thank the anonymous reviewers for their valuable suggestions. David Yuen would like to acknowledge support from Math-Geo and ITR grants from the National Science Foundation. Other funding includes NSF CAREER ACI-0238486, DMS-0528492 and the Army High Performance Computing Research Center under the auspices of the Department of the Army, Army Research Laboratory cooperative agreement number DAAD19-01-20014. We especially thank Rongjiang Wang for his assistance on simulation data generation. We also appreciate our discussions with Professor Huai Zhang. In the images of Figures 1, 10 and 11, the copyright of the geological contexts of our visualization results in the above images belongs to the Google Inc. and other geological information providers.

References

Ammon, C.J., Ji, C., Thio, H-K., Robinson, D., Ni, S., Hjorleifsdottir, V., Kanamori, H., Lay, H., Das, S., Helmberger, D., Ichinose, G., Polet, J., and Wald, D. (2005), *Rupture Process of the 2004 Sumatra-Andaman Earthquake*, Science *308*, 1133–1139.

ATI Technologies Inc. (2006), http://www.ati.com.

Buck, I. (2004), *GPGPU: General-Purpose Computation on Graphics Hardware–GPU Computation Strategies and Tricks*, ACM SIGGRAPH Course Notes (Aug. 2004).

Bilham, R. (2005), *A flying start, then a slow slip*, Science *308*, 1126–1127.

Butler, D. (2006), *Virtual globes: The web-wide world*, Nature *439*, 776–778.

Deering, M., Winner, S., Schediwy, B., Duffy, C., and Hunt, N., *The triangle processor and normal vector shader: A VLSI system for high performance graphics*. In *SIGGRAPH'88* (New York, NY, USA, ACM Press 1988) pp. 21–30.

Dias, M.L. (1991), *Ray tracing interference color*, IEEE Computer Graphics and Applications *11* (2), 54–60, March/April.

Google Earth (2007), http://earth.google.com/.

GONDEK, J.S., MEYER, G.W., and NEWMAN, J.G., *Wavelength dependent reflectance functions*. In *SIGGRAPH'94* (New York, NY, USA, ACM Press 1994) pp. 213–220.

GPGPU (2006), *General-purpose computation on GPUs*, http://www.gpgpu.org.

GUENTHER, R.D., *Modern Optics* (Wiley 1990).

KRIKKE, J. (2005), *Near Real-time tsunami computer simulations within reach*, IEEE Computer Graphics and Applications *25* (5), 16–21, Sept/Oct.

LAY, T., KANAMORI, H., AMMON, C.J., NETTLES, M., WARD, S.N., ASTER, R.C., BECK, S.L., BILEK, S.L., BRUDZINSKI, M.R., BUTLER, R., DESHON, H.R., EKSTROM, G., SATAKE, K., and SIPKIN, S. (2005), *The Great Sumatra-Andaman Earthquake of 26 December 2004*, Science *308*, 1127–1133.

LIU, Y., SANTOS, A., WANG, S.M., SHI, Y., LIU, H., and YUEN, D.A. (2007), *Tsunami hazards along Chinese coast from potential earthquakes in South China Sea*, Phys. Earth Planet. Inter. (PEPI) *163*, 233–244.

MARK, W.R., GLANVILLE, R.S., AKELEY, K., and KILGARD, M.J., *Cg: A System for Programming Graphics Hardware in a C-like Language*. In *SIGGRAPH'02* (New York, NY, USA, ACM press 2002) pp. 896—907.

MASSONNET, D., ROSSI, M., CARMONA, C., ADRAGNA, F., PELTZER, G., FEIGL, K., and RABAUTE, T. (1993), *The displacement field of the Landers earthquake mapped by radar interferometry*, Nature *364*, 138–142.

MOLNAR, S., EYLES, J., and POULTON, J. *Pixelflow: High-Speed Rendering Using Image Composition*. In *SIGGRAPH'92* (New York, NY, USA, ACM Press 1992) pp. 231–240.

NVIDIA CORP. (2004), *Deferred Shading*, http://developer.nvidia.com/.

NVIDIA CORP. (2006), http://www.nvidia.com.

NOURBAKHSH, I., SARGENT, R., WRIGHT, A., CRAMER, K., MCCLENDON, B., and JONES, M. (2006), *Mapping disaster zones*, Nature *439*, 787–788.

OKADA, Y. (1992), Internal Deformation Due to Shear and Tensile Faults in a Half-space. Bulletin of the Seismological Society of America, *82* (2), 1018–1040.

OPENGL SHADING LANGUAGE (2006), http://www.opengl.org/documentation/glsl/.

PARK, J., SONG, T.A., TROMP, J., OKAL, E., STEIN, S., ROULT, G., CLEVEDE, E., LASKE, G., KANAMORI, H., DAVIS, P., BERGER, J., BRAITENBERG, C., VAN CAMP, M., LEI, X., SUN, H., XU, H., and ROSAT, S. (2005), *Earth's free oscillations excited by the 26 December 2004 Sumatra-Andaman Earthquake*, Science *308*, 1139–1144.

SHISHKOVTSOV, O., Chapter 9. *Deferred shading in STALKER*. In *GPU Gems II: Programming Techniques for High-Performance Graphics and General-Purpose Computation* (Addison Wesley, 2nd edition 2005) pp. 143–545.

SMITS, B.E. and MEYER, G. (1994), *Newton's Colors: Simulating Interference Phenomena in Realistic Image Synthesis*. In Proc. *Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics*, pp. 185–194.

STEIN, S., and OKAL, E. (2005), *Seismology: Speed and size of the Sumatra earthquake*, Nature *434*, 581–582.

SUBARYA, C., CHLIEH, M., PRAWIRODIRDJO, L., AVOUAC, J-P., BOCK, Y., SIEH, K., MELTZNER, A.J., NATAWIDJAJA, D.H., and MCCAFFREY, R. (2006), Plate-boundary deformation associated with the Great Sumatra-Andaman earthquake, Nature *440*, 46–51.

SUN, Y., FRACCHIA, F.D., CALVERT, T.W., and DREW, M.S. (1999), *Deriving spectra from colors and rendering light interference*, IEEE Computer Graphics and Applications *19* (4), 61–67.

TITOV, V., RABINOVICH, A.B., MOFJELD, H.O., THOMSON, R.E., and GONZALEZ, F.I. (2005), *The global reach of the 26 December 2004 Sumatra tsunami*, Science *309*, 2045–2048.

USAID (2006), http://www.usaid.gov/locations/asia near east/tsunami/trm.html.

VAN PUYMBROECK, N., MICHEL, R., BINET, R., AVOUAC, J.P., and TABOURY, J. (2000), *Measuring earthquakes from optical satellite images*, Appl. Optics Inform. Processing *39* (23), 3486–3494.

VIGNY, C., SIMONS, W.J., ABU, S., BAMPHENYU, R., SATIRAPOD, C., CHOOSAKUL, N., SUBARYA, C., SOCQUET, A., OMAR, K., ABIDIN, H.Z., and AMBROSIUS, B.A.C. (2005), *Insight into the 2004 Sumatra-Andaman earthquake from GPS measurements in Southeast Asia*, Nature *436*, 201–206.

WANG, R., LORENZO-MARTIN, R., and ROTH, F. (2006), *PSGRN/PSCMP - A new code for calculating co- and post-seismic deformation, Geoid and gravity changes based on the viscoelastic-gravitational dislocation theory*, Comp. Geosci., *32* (4), 527–541.

WARD, S.N. (2006), http://es.ucsc.edu/ ward/.

WEST, M., Sachez, J.J., and McNutt, S.R. (2005), *Periodically triggered seismicity at Mount Wrangell, Alaska, after the Sumatra earthquake*, Science *308*, 1144–1146.

YOUNG, T. (1802), *A Syllabus of A Course of Lectures on Natural and Experimental Philosophy*, London: Royal Institution.

YOUNG, T. (1807), *A Course of Lectures on Natural Philosophy and the Mechanical Arts*, London: J. Johnson.

YUAN, X., LIU, Y., CHEN, B., YUEN, D.A., and PERGLER, T. (2007), *Visualization of high dynamic range data in geosciences*, Phys. Earth Planet. Inter. (PEPI) *163*, 312–320.

YUAN, X., NGUYEN, M.X., CHEN, B., and PORTER, D.H. (2005), *High dynamic range volume visualization*, Proc. IEEE *Visualization 2005*, pp. 327–334. .

YUAN, X., NGUYEN, M.X., CHEN, B., and PORTER, D.H. (2006), *HDR VolVis: High dynamic range volume visualization*, IEEE Transact. Visualization and Computer Graphics *12* (4), 433–445.

(Received November 1, 2006, revised April 10, 2007, accepted May 30, 2007)