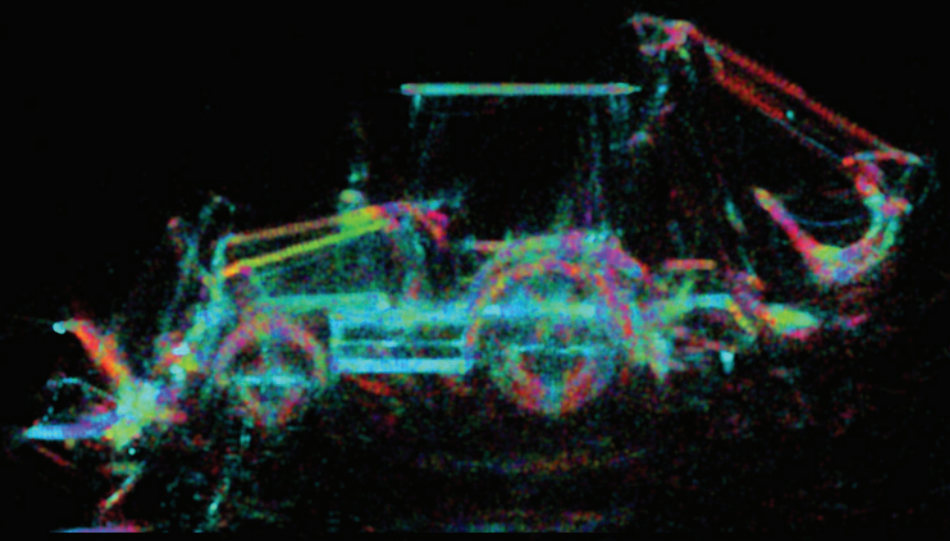
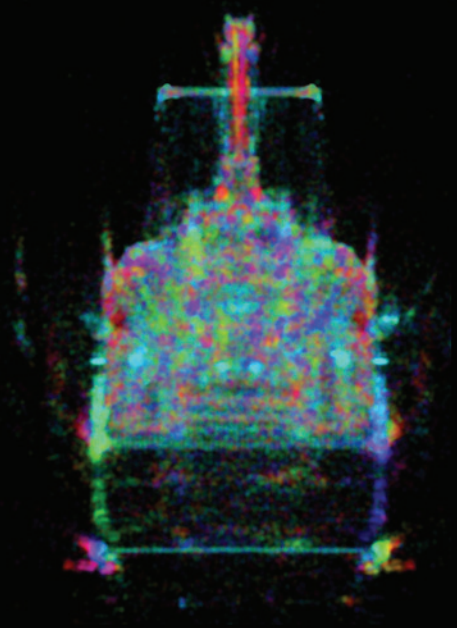
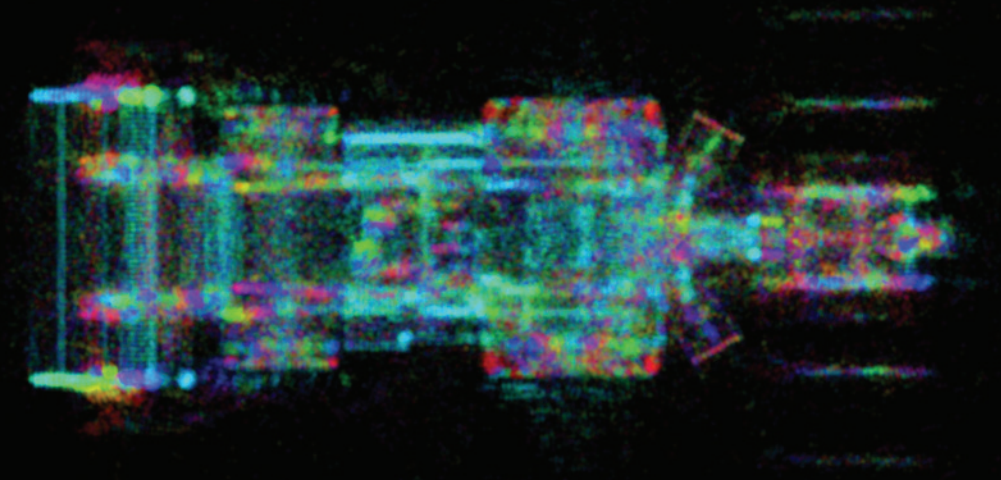
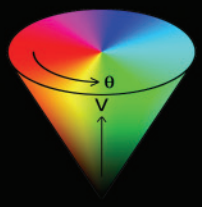
ERDC MSRC RESOURCE

Major Shared Resource Center









MSRC teams with scientists to develop new computational techniques for obtaining and visualizing 3-D reconstructions for automatic target recognition from Interferometric Synthetic Aperture Radar.

HIGH PERFORMANCE COMPUTING FOR THE WARFIGHTER. FALL 2005

from the director . . .

John West
Director, ERDC MSRC

Bill Joy—the father of Berkeley UNIX, cofounder of Sun Microsystems, and major force behind Java—recently commented in an interview that technology has matured such that we are at an inflection point in how we shape the world around us and create our future in it.

For the first time, we have begun to design things from the bottom up. For example, rather than shaping and planning and carving our way into new technology, we are designing new materials by piecing together individual atoms to create a desired set of properties.

This is a powerful shift that many have already pointed out will fundamentally change the kinds of technologies we produce as artifacts of our culture. The computer is a key tool making this shift possible.

Many of you are part of this trend. Some of you have even helped make it happen. Our job is to ensure that you continue to have the best tools available as you continue to shape our collective future. The most critical resource we provide is our staff; our computational engineers, visualization specialists, domain scientists, and support staff are the glue that joins supercomputers to the job of getting your work done. But the enabling piece that makes their work possible is the hardware itself.

As a member of the U.S. Army Engineer Research and Development Center Major Shared Resource Center (ERDC MSRC) family, I'm sure you are aware of our most recent hardware installation. As this goes to press in early fall, we are installing a 4,176-processor Cray XT3. The XT3 is an Opteron-based machine capable of providing 21.7 TFLOPS of computational capability to the user community. This machine is a tremendous resource, and we anticipate that it will enable a host of new and expanded applications at a size and fidelity not previously attainable on Department of Defense (DoD) resources.

We are also proud to announce that this machine is already setting world records. The ERDC XT3, named sapphire (read more about it inside this issue), is the most capable machine listed on the High Performance Computing (HPC) Challenge benchmark suite in two important tests: the random access and FFTE measurements. The news gets even better in that we are still in the early stages of configuring this machine, and the expectation is that we will be setting even more records in the coming months.

You may be asking why I am talking about the HPC Challenge benchmark suite instead of the more traditional LINPACK measurement used for the popular Top500 list. Both are important, and we will be talking about Top500 performance in the coming weeks as well. But the HPC Challenge suite has something the HPC community and our Nation's leaders have been missing for a long time in framing the HPC debate: depth.

Today's high performance computers are complex machines, and the workload served by the computers is tremendously varied. Rank on the Top500 list is determined by performance on a single computationally intensive dense matrix kernel from LINPACK. This metric, while important, is indicative of only one of the several dimensions of a machine's architecture that affect its suitability for any given computational problem. In contrast, the HPC Challenge suite provides the community with a set of performance measures that come closer to spanning the set of machine features of interest to today's users without overwhelming the community with a vast array of data. In short, HPC Challenge results illuminate more facets of a machine's performance than a single number can.

Why does this matter? Policy makers are busy folks with a lot of issues to consider. To quickly address each different issue, they need a common vocabulary on which to frame the national debate. Pick any issue, and you will find that our top leaders do their jobs effectively only insofar as the vocabulary they select allows them to accurately and completely express the nature of the issue with which they are dealing. For decades a viable alternative to the Top500 list has not existed, with

About the Cover: Cover designs by the ERDC MSRC Scientific Visualization Center. For related stories, see page 3 (front cover) and page 15 (back cover).

the result that policy and, to a lesser extent, acquisition decisions have been driven by who did and did not perform well on one measure of that list. The end result is an HPC ecosystem that is less robust than it otherwise might have been.

Vendors are ultimately driven to design what a market will buy. My hope is that as leaders in the HPC community, we can begin to talk routinely about more of the dimensions of performance. In so doing, we can educate our policy and decision makers about the complex and sometimes tempestuous marriage between computational problems of national interest and the high performance computers used to solve those problems. The HPC Challenge suite provides us a new vocabulary with which to frame a more interesting debate. Whether we can do this without losing our audience in the added complexity will be an important test for the leadership of our community.

As always, I want to hear from you! If you'd like to let me know how we're doing, share a success story, or make a suggestion for ways to improve our service, drop me a line at **john.e.west@erdc.usace.army.mil**.

John E. West, Director
Major Shared Resource Center
U.S. Army Engineer Research and Development Center
Vicksburg, MS

Contents

from the director . . .

Three-Dimensional Target Visualization from Wide-Angle IFSAR Data
By Randolph L. Moses, The Ohio State University, and Paul Adams, Randall Hand, and Tommy Biddlecome, ERDC MSRC 3

ERDC MSRC Installs Most Powerful Supercomputer in DoD
By Jay Cliburn 10

solutionHPC 11

Technical Overview of ezHPC Architecture
By Weston Monceaux, Keith Rappold, and Patti Duett 12

ezVIZ – An Open-Source, Cross-Platform Visualization Tool
By Randall Hand and Paul Adams 15

About PET
By Dr. Jim Pennington 20

Storage Monitoring and Forecasting at the ERDC MSRC
By Teresa Moncrief and David Sanders 24

DoD Supercomputing Center at ERDC to be Featured Worldwide Once Again
By Rose J. Dykes 26

Helping Shape the Future 27

ERDC MSRC Makes 11 Presentations at UGC 2005 and Brings Home Two Hero Awards
By Rose J. Dykes 28

visitors *29*

acronyms *31*

training schedule *32*

Three-Dimensional Target Visualization from Wide-Angle IFSAR Data

By Randolph L. Moses, The Ohio State University, and Paul Adams, Randall Hand, and Tommy Biddlecome, ERDC MSRC

Introduction

This article considers the problem of developing three-dimensional (3-D) spatial representations of objects by processing radar measurements of that object. The measurements that are wideband (have high-range resolution) and that are collected across wide-angular apertures in both azimuth and elevation are of particular interest. The goal is to provide 3-D spatial reconstructions that can be used first to visualize the object along with radar features that characterize the object (such as polarization features) and, later, can be used in automatic target-recognition applications. Thus, visualization provides a mechanism for human-aided target recognition and also serves as a qualitative means for demonstrating the utility of extracted feature sets for automatic target recognition.

Traditional approaches for 3-D synthetic aperture radar (SAR) processing form a 3-D spatial reconstruction from a dense set of measurements over a frequency-azimuth-elevation data cube. While such an approach has several merits, it also has some disadvantages. First, the measurement and processing requirements are significant. For example, to obtain a 3-D SAR image with spatial resolution of 0.15 m in each dimension requires phase-coherent collection of data over approximately 1.2 GHz (in frequency) and over a 20° × 20° azimuth-elevation sector. In addition, coherent processing of this data cube imposes significant memory and computational demands. Another disadvantage of traditional SAR imaging techniques is that they implicitly assume that scattering responses persist across the measurement aperture, an assumption that becomes tenuous for wide-aperture measurements. Thus, even if the significant challenges of data collection, storage, and processing overfilled wide-angle apertures could be overcome, the processing of data deserves reconsideration in light of this scattering persistence assumption.

This article considers an alternative approach for forming 3-D reconstructions, one that uses a sparse set of measurements over wide angles. In the proposed approach, measurements are collected as a set of high-resolution interferometric SAR (IFSAR) image pairs that are distributed over wide angles. Each image pair is coherently processed to estimate 3-D scattering locations, and associated scattering attributes, over a limited aperture. These attributed points are combined *noncoherently* to form the 3-D reconstructions. The noncoherent combination is significant because it relaxes the requirement for phase coherence across wide-angle apertures; in particular, multiple radar measurement platforms operating incoherently from one another are able to collect data for this reconstruction.

The proposed IFSAR processing procedure results in a set of 3-D spatial point locations of scattering centers. Associated with each scattering center is a vector of real-valued features, or attributes. Analysis and visualization of this high-dimensional data represent a second significant challenge. Volume rendering results in which volume clouds with varying colors and transparencies are used to represent the data.

The processing and visualization approach is illustrated using synthetic, wide-angle scattering data of a ground vehicle, a backhoe. Results are presented in which scattering amplitude, polarization parameters, and observation-angle parameters are represented in the reconstructions. The reconstructions indicate the power of the proposed visualization techniques for understanding high-dimensional point data.

Wide-Angle Radar Scattering

The goal is to develop 3-D reconstructions of objects from radar backscatter measurements that span a possible wide range of frequencies, azimuths, elevations, and, possibly, multiple polarizations. These (monostatic) backscatter measurements can be written as the response

$$S(f, \varphi, \psi) = \begin{bmatrix} S_{HH}(f, \varphi, \psi) & S_{HV}(f, \varphi, \psi) \\ S_{VH}(f, \varphi, \psi) & S_{VV}(f, \varphi, \psi) \end{bmatrix} \quad (1)$$

where f is measurement frequency and where (φ, ψ) denotes the azimuth and elevation angle (with respect to the object) that the (far-field) measurement is taken. The subscripts in Equation 1 denote the receive and transmit polarization of the measurement.

Reconstruction is the process of transforming backscattering measurements to a 3-D spatial representation of the object. Typical reconstruction methods include SAR image formation in which a set of 2-D measurements in a frequency-angle interval (e.g., $[f_L, f_H] \times [\varphi_L, \varphi_H]$ for a fixed ψ) is used to form an image that represents radar backscatter as a function of two spatial dimensions in an

image plane using tomographic processing techniques. Three-dimensional SAR is a generalization in which the data are a "cube" in (f, φ, ψ)-space, and a 3-D spatial-backscattering representation is obtained. Both methods typically employ tomographic processing and, thus, implicitly assume that the scattering response at the spatial location (x, y, z) is constant over the (f, φ)- or (f, φ, ψ)-measurement interval. This constant-response assumption can be a good assumption for narrow-percent-bandwidth frequency intervals and narrow-angular intervals. However, for wide-percent-bandwidth measurements, and for wide-interrogation-angle widths in either φ or ψ, scattering can exhibit significant variation as a function of frequency and angle.[1] For such cases, traditional SAR image or volume reconstruction processing should be reconsidered.[2, 3]

In general, the backscattered response at location (x, y, z) on the object depends on frequency, angle, and polarization; this response can be thought of as an "antenna pattern" whose amplitude and phase depend on both angle (azimuth and elevation) and polarization. Thus, each location in the reconstruction volume is characterized by a set of features describing the polarimetric response as a function of angles φ and ψ. In addition, the angle-dependent and polarization-dependent properties of scattering are strongly related to the local shape of the scattering object; in particular, curvature and orientation determine the scattering behavior.[1] As a result, 3-D reconstructions that incorporate and represent these features can provide strong indications to object shape.

A complete 3-D reconstruction would, in general, require coherently measuring the object over all frequencies and angles; this results in a measurement "data dome" in frequency-aspect space (or k-space) shown in Figure 1. This represents an enormous amount of data and poses significant challenges both in the acquisition and processing of the data. Thus, 3-D reconstruction methods are sought based on sparse subsets of the data dome.

Sparse Aperture 3-D Reconstruction

This article considers a sparse reconstruction technique in which both the data collection and data processing are simplified by exploiting properties of radar-scattering physics. These properties are discussed below.

Limited persistence of scattering

Since object-scattering behavior over wide angles is not well-modeled as independent of angle, small-angle scattering assumptions that form the basis of most current object reconstruction methods are no longer

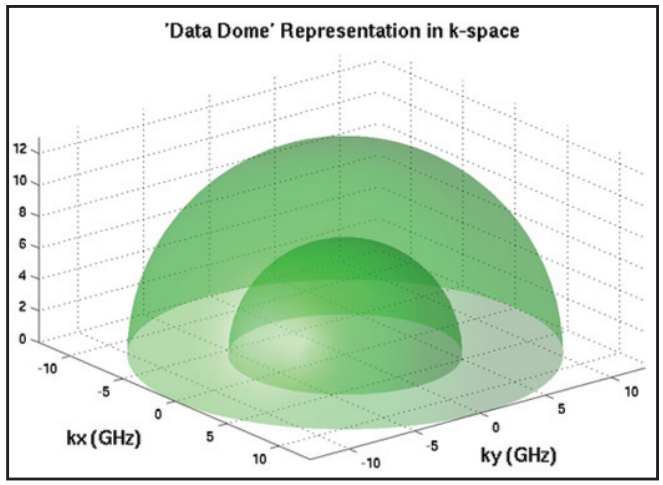


Figure 1. Data dome of frequency-angle measurements for complete object reconstruction

valid. Most scattered energy from objects of interest has limited angles of persistence and motivates developing reconstruction algorithms that accommodate limited persistence.

In particular, limited persistence suggests a *noncoherent* combination of information from subapertures of wide-angle measurements.[2,3]

Scattering information is extracted from limited-angle SAR images, where the angle extent of the data used to form the image is matched to the persistence angle of scattering. Recently developed methods are adopted for a noncoherent combination of features extracted from coherent subaperture imaging,[2,3] where subapertures are matched to scattering persistence angles.

Isolated, dominant scattering centers

At high frequency, radar backscattering from objects is well-modeled as a sum response from individual *scattering-centers*.[1] Also, at sufficiently high resolutions, scattering centers on the object become resolved, and resolved scattering center responses have both stable-phase responses and stable polarimetric-response properties.[3,4] As a result, 3-D location information inferred from the scattering phase is sufficiently stable to permit 3-D object reconstructions using IFSAR techniques. Limited persistence of scattering is no longer dominated by scintillation effects and, instead, relates to object shape. Furthermore, polarimetric features extracted from scattering centers are useful parameters to describe local geometry of the object.

Polarimetric, wide-angle IFSAR processing for 3-D reconstruction

Based on the above considerations, the authors of this article propose a wide-angle object reconstruction that employs IFSAR processing to arrive at 3-D spatial

scattering from processing of limited-angle apertures, along with noncoherent combinations of these points over wider angle apertures. A summary of the approach follows. Necessary data are collected to form a pair of coherent SAR images at two closely spaced elevation angles; the frequency and azimuth extent of this data set pair are sufficiently large to obtain high-resolution images (e.g., 2 in. × 2 in.) in which most scattering centers on the images are isolated. Standard image formation is used, so scattering amplitude is implicitly presumed to be persistent across the measurement aperture. From each pair of images, which is referred to as an IFSAR pair, estimates are made for the 3-D location of scattered energy corresponding to high-amplitude image pixels; and the height of the scattering off the image plane is computed from the phase difference of corresponding pixels from the two images, using standard IFSAR processing methods.[5] Each 3-D point has a number of attributes, including polarization features and the center azimuth and elevation angle for measurements used to form this IFSAR image pair. This process is repeated for several IFSAR image pairs whose center angles span a wide range. Finally, the 3-D points are noncoherently combined, spatially smoothed, and rendered for visualization. The resulting data are of high dimension, and subsets of the data are rendered using color and transparency to visualize selected attributes.

IFSAR processing

Each IFSAR image pair is processed as follows. All image pixels whose radar cross section (RCS) is above a fixed threshold are determined. If single-polarization imagery is available, the RCS is computed as the pixel magnitude; when two or more polarizations are measured, the RCS is computed as the root-mean-square (RMS) sum of pixel magnitudes over the available polarizations. For each pixel exceeding the RCS threshold, its slant-plane downrange and crossrange location (x_s, y_s) is taken as the pixel center, and its height is computed from the slant plane using the difference in phase between the pixel value from the SAR image centered at (φ_k, ψ_k + 0.05°) and the image centered at (φ_k, ψ_k):

$$z_s = \frac{\lambda}{4\pi\Delta\psi}\left[\angle s_2 - \angle s_1\right] \tag{2}$$

where l is the radar wavelength at the center frequency, $\Delta\psi$ is the elevation difference angle of the IFSAR pair (0.05° in this case), and s_1 and s_2 are the pixel values from the two images. This 3-D location is transformed from the local slant-plane coordinate system to a coordinate (*x, y, z*) in an absolute, target-centered coordinate system, referred to as an "IFSAR point." Also, a test to determine whether $|s_1| \approx |s_2|$ and points whose relative amplitudes are dissimilar are rejected, since amplitude dissimilarity indicates that the pixel values may not result from a single dominant scattering term.[6] Several features are associated with each IFSAR point: the center aspect (jk, y$_k$) of the IFSAR image pair and the (possibly polarimetric) response amplitude. Thus, each IFSAR point is attributed with 10 real-valued features in the fully polarimetric case.

Polarimetric features

The response amplitude may be a single RCS value or a polarimetric characterization of amplitude. If full-polarization measurements are available, the amplitude is characterized by the 2 × 2 complex-valued matrix of scattering amplitudes from each of the four component polarizations. A Pauli-basis decomposition of this scattering matrix is formed.[7]

$$[S] = A\,[\cos(\alpha)[S]^t + e^{j\varphi}\sin(\alpha)_\cos(\theta)[S]^{\mathrm{d}}_0 + \sin(\theta)[S]^{\mathrm{d}}_{45}] \tag{3}$$

where 0° ≤ α ≤ 90°, -45° ≤ θ ≤ 45°, and where the Surface-Dihedral-Tilted Dihedral basis is

$$[S]^t = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \quad [S]^d_0 = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \quad [S]^d_{45} = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

The matrix $[S]^t$ represents an ideal trihedral (or other odd-bounce scattering object, such as a flat plate or a sphere); $[S]^d_0$ represents a horizontally oriented dihedral (an even-bounce scattering term); and $[S]^d_{45}$ represents a diagonally oriented dihedral (also even-bounce). Thus, the decomposition in Equation 3 expresses the scattering matrix as a decomposition of these three canonical terms. The real parameters A, α, and θ, in particular, provide information about the physical characteristics of the dominant scattering mechanism in the resolution cell. The parameter α represents the mixture fraction of the trihedral and dihedral components. For α = 0°, the scattering center is entirely a trihedral component, and for α = 90°, it is entirely a dihedral. The angle θ gives the orientation angle of the dihedral term. A represents the overall RCS of the scattering center. To summarize, IFSAR processing provides (*x, y, z*) locations of scattering centers that are persistent about a region centered at azimuth and elevation (φ_k, ψ_k). Scattering RCS A and polarimetric features such as α and θ further characterize the points. Thus, the radar signal processing output is a set of (*x, y, z*) locations, each attributed with several real-valued feature parameters. The next section considers ways of visualizing this high-dimensional point data representation.

Visualization Techniques

Each data set consists of a multiscalar, rectilinear brick of floats with a spatial resolution of $1{,}024^3$. A commonly prevalent visualization technique would be to use isosurface generation to rebuild and view surfaces contained in the data. However, using isosurfacing techniques in a data set containing sparse or weakly defined structures often results in inaccurate surface representations. The problems of such surface reconstructions are related to making a decision for each volume element, whether or not the surface passes through the element. Surface-rendering techniques in sparse and noisy data sets with weakly defined features would produce both false positives (spurious surfaces) and false negatives (erroneous holes) resulting in inaccurate representations of the true features of interest. Instead, two different, but complementary, volume techniques produce more positive results in exploration and visualization of these data sets.

Volume rendering

Volume rendering algorithms[8,9,10] can directly display the volume information contained in the data set through semitransparent images. This allows the user to explore the internal structures of the image volume without reconstruction of intermediate geometrical representations--thus eliminating the requirements to algorithmically decide where a surface exists. This provides the ability to display volumes containing both noise and faint surfaces and to overcome the problems of inaccurate representation of surfaces inherent in the surface-rendering techniques. A volume is defined as a 3-D array of voxels in an intensity (scalar) field, and a color volume (or red green blue alpha (RGBA) volume) to be a 3-D array of voxels in and RGBA (color and opacity) field. Color and opacity are assigned to the scalar values of each voxel of the data set. Volume rendering involves the forming of an RGBA volume from the data, reconstruction of a continuous function from this discrete data set, and projecting it onto a 2-D viewing plane from a desired perspective.

A key component in RCS data set volume visualization is the transfer function that maps the volume's intensity values to color and opacity values for display. In surface rendering,[11] the transfer function defines the thresholds with which isosurfaces are extracted and rendered as surface objects. In volume rendering, the transfer function is used as both a filter that selects a subset of the volume information to be presented and as an information interpreter that determines how the selected information is to be displayed. Selection of a proper transfer function is crucial, as it defines what is displayed in the generated image.

The transfer function is defined as a two-step process. The first step consists of sequence of intensity mappings in a volume's intensity field and performs tasks such as information filtering and noise reduction. The second step is essentially the "*coloring*" process, which generates colors and opacity values directly from intensity values using an intensity-to-RGBA color look-up table. In the color look-up table (Figure 2), a linear ramp is used for the color components, since the necessary "intensity processing" is done in the first step. By using a transfer function, each voxel sample is shaded and projected onto a plane to form images. Typically, a volume-rendering technique has been applied to extracting views of a surface or set of surfaces. In this case, however, the data are not represented in terms of a voxel. Instead, they are represented as point data that can have multiple scalars assigned to them. A rectilinear grid was created and the point data interpolated onto it. Each voxel could potentially contain a multiple number of points, each with its multiple scalar values. For example (Figure 2), the point data with the highest RCS value is selected to reside within each voxel. An appropriate transfer function based on the RCS value was then applied. Lower RCS-valued voxels had a lower amount of opacity. Higher RCS-valued voxels had a higher amount of opacity. The output image in this case represents an RCS-intensity view of the data.

For volume rendering, the Visualization Toolkit (VTK), an open-source program from Kitware, was used to render the data. In addition, applying a volume-based technique to point-based data was felt to be inappropriate. A second, point-based approach was then used.

Point rendering (point clouds)

Point-based representations (PBRs) may also define a geometric shape by rendering a dense cloud of sampled points on the surface. PBRs allow for efficient and flexible handling of highly detailed 3-D models like the ones generated by high-resolution SAR devices. Levoy and Whitted[12] initially proposed the use of points for rendering objects, and the technique was expanded on by other researchers,[14,15,16] such that points can be rendered directly, achieving a comparable effect as the previously described volume rendering. At first analysis, one would think PBRs would render faster than using polygonal objects since points do not have the associative information needed for surface rendering such as face, connectivity, or normals. However, the numbers of

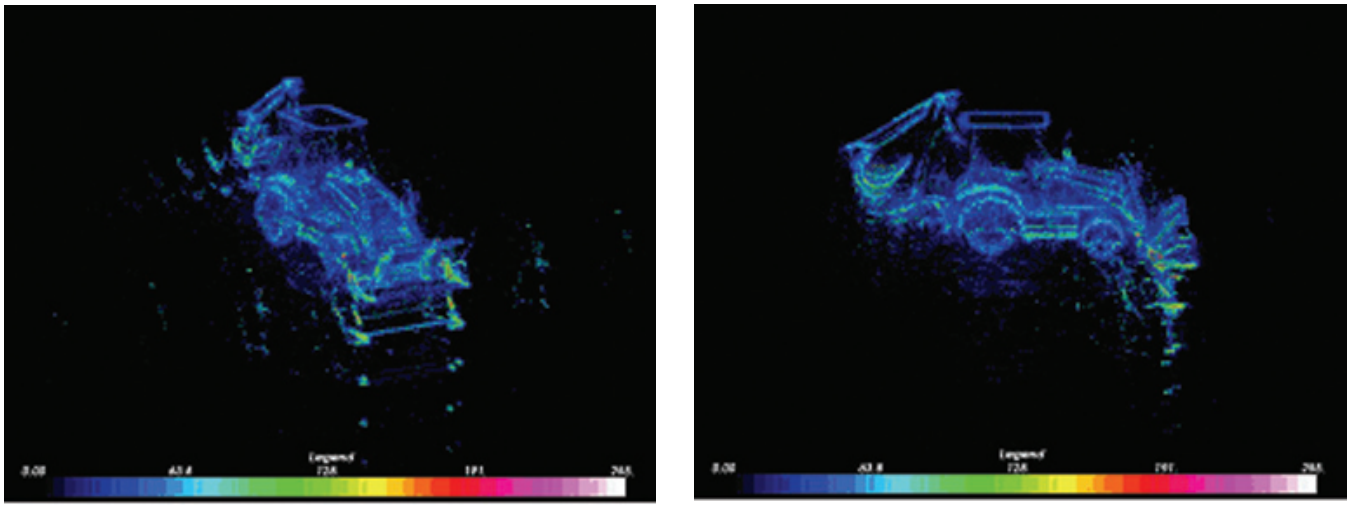


Figure 2. Volume rendering via ray casting with color representing RCS intensity

samples needed to define a surface or object are typically much greater than for polygonal surfaces. Point-rendering complexity was described by Zwicker et al.[13] using high-quality point rendering, where points as fundamental elements consisted of position, color, radius, etc. Botsch et al.[17] determined that one trades some rendering performance for rendering quality.

With the volume-rendering technique, point data were interpolated onto a rectilinear grid, and then the transfer function was applied to generate an image. With the point cloud technique, the transfer function was applied directly to the points. The data set was rendered in a 3-D conceptual package (Maya) as a point-based representation, or a cloud. Opacity was applied to the points based on their RCS value. Singular points (or noise) are rendered but tend to disappear because of the size of the point and its transparency allowing groups of points to show up more readily. Color for each point was applied using three transfer functions to highlight different scalar attributes.

Backhoe Reconstruction Examples

This section presents 3-D representation and visualization results using the proposed sparse aperture reconstruction techniques applied to synthetic monostatic backscatter predictions of a backhoe.

Wide-angle backhoe data

The data used in this study consist of scattering predictions of a backhoe shown in Figure 3.

A facet model of the backhoe, along with the Xpatch scattering prediction program, is used to generate fully polarimetric, monostatic backscattering predictions for frequencies between 7 and 13 GHz. This study considers a sparse sampling of the full data dome, as follows. XpatchT is used to generate SAR slant plane images

Figure 3. Backhoe facet model used in Xpatch scattering predictions

with resolution 2 in. × 2 in. SAR image pairs are synthesized, where the kth image pair has center azimuth and elevation at (φ_k, ψ_k) and $(\varphi_k, \psi_k + 0.05°)$ and where

$$\varphi_k \in [0°, 5°, \ldots, 355°]; \psi_k \in [0°, 5°, \ldots, 85°] \quad (4)$$

Each image thus requires backscattering measurements over a frequency band of approximately 8 to 12 GHz and an angle span of approximately 24° centered at its particular (φ, ψ) angle pair. For processing, two images in each SAR image pair are required to be phase-coherent with one another, but phase coherence is not required from one IFSAR pair to another. This means, for example, that separate radar platforms can measure SAR images with different (φ_k, ψ_k) center angles, and these platforms need not be phase-coherent with one another. This section presents several reconstructions of the backhoe. These reconstructions illustrate the

utility of some of the features associated with the 3-D scattering points.

Backhoe 3-D reconstruction results

Three reconstruction examples are presented that illustrate visualization techniques that use three subsets of the five point-scattering features described earlier.

Figure 4 shows three sample views of a 3-D backhoe reconstruction. In this reconstruction, only the scattering RCS feature is shown. In this figure, the RCS value is encoded by color and voxel transparency. The reconstruction shows significant fine detail. The noncoherent combination of points provides a reconstruction filled from all angles (even though individual scattering terms may persist over much smaller angles) that are more reminiscent of optical imagery and traditional radar imagery.

Figure 5 shows three sample views of the backhoe, where color is used to depict polarization properties. In this image, color encodes the value of the α parameter, where $\alpha = 0°$ is green (odd-bounce) and $\alpha = 90°$ is blue (even-bounce or dihedrals); orange represents mixtures of the two. Nearly all scattering is seen to be either mostly odd-bounce or mostly even-bounce, and only a small percentage of points show a mixture. The correspondence between the alpha color and local geometry of the backhoe is strong and matches well with the expectations from target geometry and associated scattering physics.

Figure 6 shows three sample views of the backhoe, where, in this case, color is used to depict the azimuth angle φ_k at which the scattering center is detected properties. Object substructures are seen to often have common color, indicating that aspect angle may be useful as a feature for grouping scattering points into substructures.

Conclusions

This article presented a method for obtaining and visualizing 3-D reconstructions of ground vehicles from high-frequency, wide-angle measurements of monostatic backscattering. Full reconstruction requires measurements over a dense grid in frequency-azimuth-elevation space, which imposes significant challenges for both data collection and data processing. The proposed method exploits properties of scattering physics to develop a reconstruction method that uses sparse aperture measurements; specifically, measurements are collected as a set of high-resolution IFSAR image pairs that are distributed over wide angles.

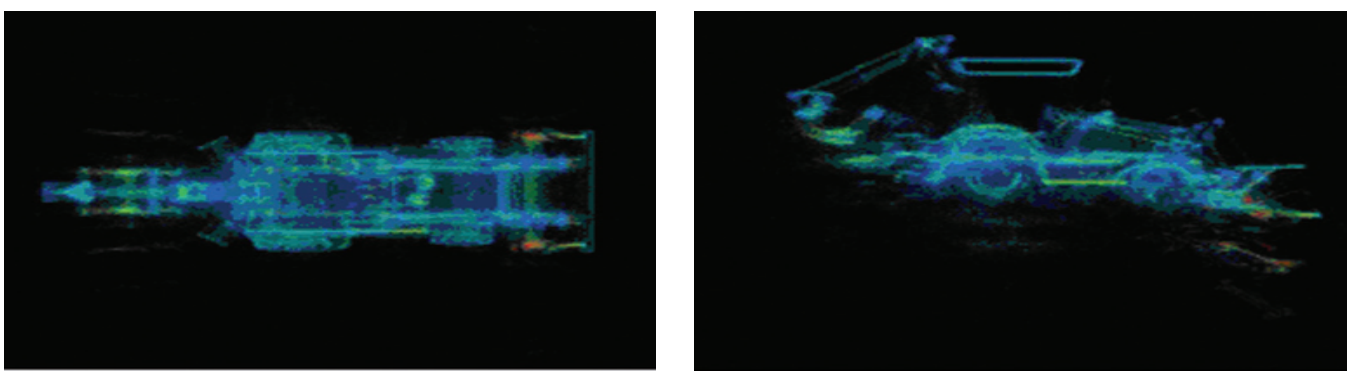


Figure 4. Volume cloud image of backhoe with color representing RCS magnitude

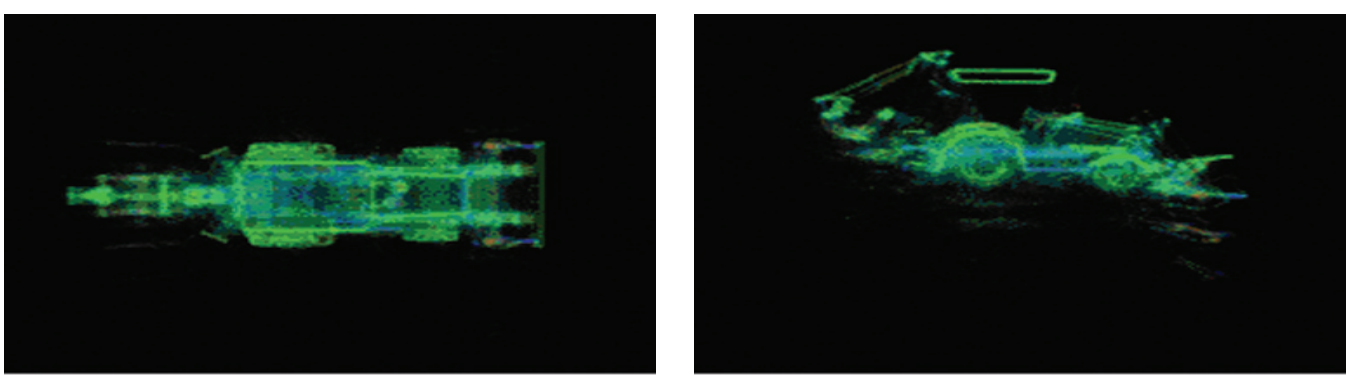


Figure 5. Volume cloud image of the backhoe with color representing the polarization parameter α, where (odd-bounce scattering) is green, $\alpha = 90°$ (even-bounce scattering) is blue, and values in between are orange

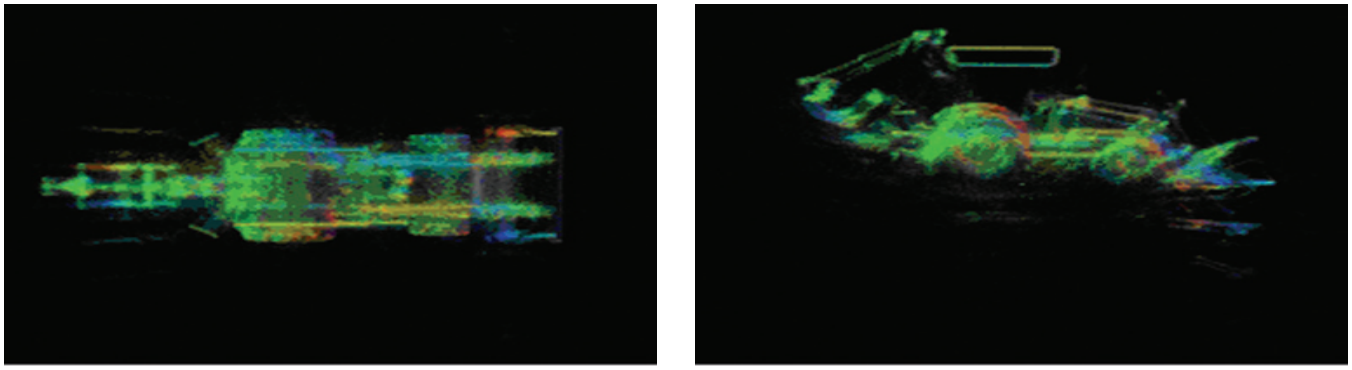


Figure 6. Volume cloud image of the backhoe with color representing the azimuth angle φ_k, associated with the scattering point

The IFSAR measurement samples are approximately 1/50 the full data dome. Moreover, each image pair is processed coherently, but the results are combined noncoherently to form the 3-D reconstructions. The noncoherent combination is significant because it relaxes the requirement for phase coherence across wide-angle apertures; in particular, multiple radar measurement platforms operating incoherently from one another are able to collect data for this reconstruction. The IFSAR processing results in a set of 3-D spatial point locations of scattering centers, each attributed by several real-valued features. Analysis and visualization of this high-dimensional data represent a second significant challenge. Volume-rendering results are presented in which volume clouds with varying colors and transparencies are used to represent the data. Results are presented in which scattering amplitude, polarization parameters, and observation-angle parameters are represented in the reconstructions. The reconstructions indicate the power of the proposed visualization techniques for understanding high-dimensional point data. In particular, they provide a qualitative analysis and comparison technique for assessing the utility of these radar features for target recognition, structure extraction, and classification

References

1. L.C. Potter and R.L. Moses, "Attributed scattering centers for SAR ATR," *IEEE Transactions on Image Processing* 6, pp. 79–91, January 1997.
2. R. Moses, L. Potter, and M.C. Etin, "Wide angle SAR imaging," in *Algorithms for Synthetic Aperture Radar Imagery XI (Proc. SPIE Vol. 5427)*, E. G. Zelnio, ed., April 2004.
3. R. Moses, E. Ertin, and C. Austin, "Synthetic aperture radar visualization," in *Proceedings of the 38th Asilomar Conference on Signals, Systems, and Computers*, (Pacific Grove, CA), November 7–10 2004.
4. Y. Akyildiz, "Feature extraction from synthetic aperture radar imagery," Master's thesis, The Ohio State University, August 2000.
5. C.V. Jakowatz, D.E. Wahl, and P.H. Eichel, *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach*, Kluwer Academic Publishers, Boston, MA, 1996.
6. C. Austin and R. Moses, "IFSAR processing for 3-D target reconstruction," in *Algorithms for Synthetic Aperture Radar Imagery XII (Proc. SPIE Vol. 5808)*, E.G. Zelnio, ed., March 2005.
7. E. Ertin and L.C. Potter, "Polarimetric classification of scattering centers," *IEEE Trans. on AES*, July 2000.
8. Marc Levoy, "Display of surfaces from volume data." *IEEE Computer Graphics and Application*, 8(3):29-37.
9. J. Danskin and P. Hanrahan, "Fast algorithms for volume ray tracing," in *Proc. 1992 Workshop on Volume Visualization*, pages 91-98, October 1992.
10. C. Upson and M. Keeler, "V-buffer: Visible volume rendering," *Computer Graphics, SIGGRAPH'88*, 22(4):59-64, August 1988.
11. W.E. Lorensen and H.E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *Computer Graphics, SIGGRAPH'87*, 21(4):163-169, July 1987.
12. M. Levoy and T. Whitted, "The Use of Points as a Display Primitive." *TR 85-022. Univ of North Carolina at Chapel Hill*, 1985.
13. H. Pfister, M. Zwicker, J. van Baar, and M. Gross, "Surfels: surface elements as rendering primitives," *Computer Graphics (Proc. SIGGRAPH 2000)*, pages 335-342. ACM Press, New York, 2000.
14. J. Grossman, "Point sample rendering," Master's thesis, Department of EE and CS, MIT, 1988.
15. S. Rusinkiewicz, M. Levoy, "QSplat: a multiresolution point rendering system for large meshes," *Computer Graphics, (SIGGRAPH 2000 Proceedings)*, pp. 343-352.
16. Y. Yemez. F. Schmitt, "Progressive multilevel meshes from octree particles," I*EEE proceedings: 3D digital imaging and modeling*, 1999.
17. M. Botsch, A. Wiratanaya, L. Kobbelt, "Efficient High Quality Rendering of Point Sampled Geometry," *Thirteenth Eurographics Workshop on Rendering* (2002).

ERDC MSRC Installs Most Powerful Supercomputer in DoD

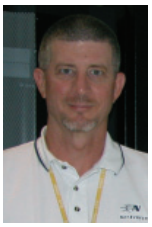
By Jay Cliburn

The ERDC MSRC is pleased to announce the installation of its latest supercomputer, a 4,176-processor Cray XT3. The system, hostname sapphire, provides a whopping 21.7 TFLOPS of peak computational power to the DoD High Performance Computing Modernization Program (HPCMP) user community and is the most powerful computer in the entire DoD. It currently stands as the most powerful computer in the *world* when measured by two of four system-wide High Performance Computing Challenge (HPPC) benchmarks, holding the world record for the Global Random Access benchmark and the Global FFTE benchmark. See http://icl.cs.utk.edu/hpcc for details and results of the HPCC benchmark. Sapphire is expected to be available for production work in the first calendar quarter of 2006.

The system is housed in 44 cabinets and comprised of two types of nodes: compute nodes and service nodes. Each compute node contains four 2.6-GHz AMD Opteron central processing units (CPUs), 4 GBytes of RAM, and runs a Catamount lightweight operating system kernel optimized for number crunching. The 1,032 compute nodes comprise a total of 4,128 compute processors. Each service node contains one Opteron CPU, 4 GBytes of RAM, and runs Suse Linux as its operating system. Service nodes are used primarily for user logins, disk input/output (I/O), and other system functions. Sapphire contains 48 service nodes, with 11 of those nodes configured for user logins. Users submit jobs from service nodes to compute nodes using the Load Sharing Facility (LSF) batch queuing system.

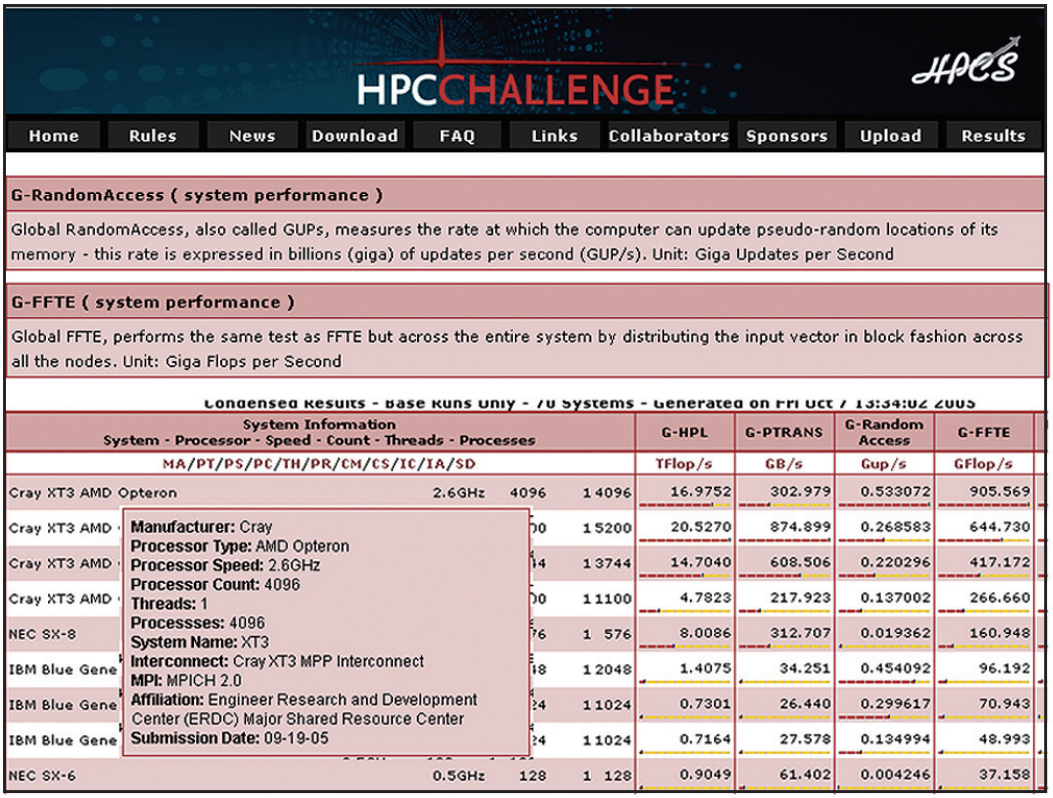
CPU-to-CPU interconnection on Sapphire is accomplished through the Cray "SeaStar" ASIC configured in a three-dimensional (3-D) torus arrangement. One SeaStar chip is allocated to each CPU, and each SeaStar can connect to six other SeaStars at a rate of 7.6 GBytes/s each. MPI and SHMEM are provided for message passing. All nodes share a common 112-TB RAID storage system configured with the Lustre parallel file system.

Sapphire is air-cooled. During the spring and summer of 2005, the ERDC MSRC upgraded the computing facility to accommodate the system by heavily modifying the cooling infrastructure, including the installation of a 300-ton water chiller to supplement the existing chilled water production facility, and nine 60-ton, 17,100-cfm air handlers to provide cooling air to the system.



Jay Cliburn
Technical Operations Manager, ERDC MSRC

The Cray XT3 brings a new level of computational power to the ERDC MSRC and to the HPCMP user community. New and exciting work products are anticipated from the talented users, enabled by the additional computing capability provided by Sapphire. As always, one should contact the HPC Service Center with questions or comments.



HPCCHALLENGE

Home | Rules | News | Download | FAQ | Links | Collaborators | Sponsors | Upload | Results

G-RandomAccess (system performance)

Global RandomAccess, also called GUPs, measures the rate at which the computer can update pseudo-random locations of its memory - this rate is expressed in billions (giga) of updates per second (GUP/s). Unit: Giga Updates per Second

G-FFTE (system performance)

Global FFTE, performs the same test as FFTE but across the entire system by distributing the input vector in block fashion across all the nodes. Unit: Giga Flops per Second

Condensed Results - Base Runs Only - 70 Systems - Generated on Fri Oct 7 13:34:02 2005

System Information System - Processor - Speed - Count - Threads - Processes					G-HPL	G-PTRANS	G-Random Access	G-FFTE
MA/PT/PS/PC/TH/PR/CM/CS/IC/IA/SD					TFlop/s	GB/s	Gup/s	GFlop/s
Cray XT3 AMD Opteron		2.6GHz	4096	1 4096	16.9752	302.979	0.533072	905.569
Cray XT3 AMD			00	1 5200	20.5270	874.899	0.268583	644.730
Cray XT3 AMD			14	1 3744	14.7040	608.506	0.220296	417.172
Cray XT3 AMD			00	1 1100	4.7823	217.923	0.137002	266.660
NEC SX-8			76	1 576	8.0086	312.707	0.019362	160.948
IBM Blue Gene			18	1 2048	1.4075	34.251	0.454092	96.192
IBM Blue Gene			24	1 1024	0.7301	26.440	0.299617	70.943
IBM Blue Gene			24	1 1024	0.7164	27.578	0.134994	48.993
NEC SX-6		0.5GHz	128	1 128	0.9049	61.402	0.004246	37.158

Manufacturer: Cray
Processor Type: AMD Opteron
Processor Speed: 2.6GHz
Processor Count: 4096
Threads: 1
Processses: 4096
System Name: XT3
Interconnect: Cray XT3 MPP Interconnect
MPI: MPICH 2.0
Affiliation: Engineer Research and Development Center (ERDC) Major Shared Resource Center
Submission Date: 09-19-05

solutionHPC

For years a growing demand has existed within the HPC community for someone to lead an effort that would allow easier access to HPC resources for novice and intermediate users. As user systems became more robust, the cry grew to include access to visualization resources.

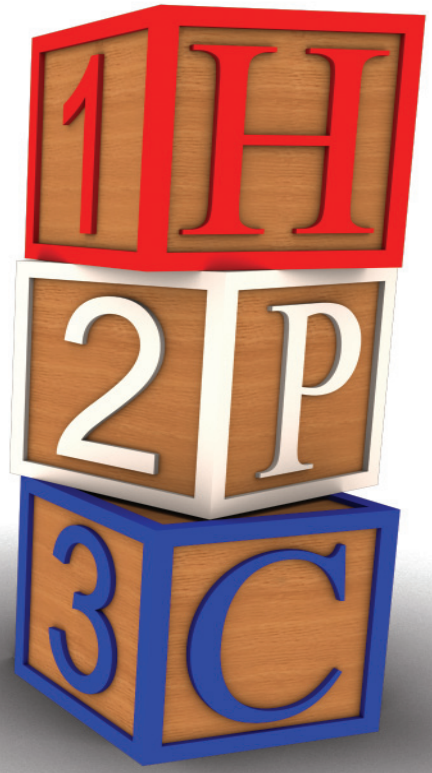
The ERDC MSRC took up this challenge – on both fronts with a project called solutionHPC. A little over a year ago, development began on the first two solutionHPC products that are designed to fill the "easy" void.

The first, ezHPC, takes a graphical user interface (GUI) approach to HPC access. The user-friendly interface is available in both .Net and Web clients. From user authentication to return of the end product back to the user's computer, ezHPC delivers a seamless point and click environment. New and power users alike will find ezHPC an attractive alternative to traditional HPC access.

The second, ezVIZ, opens the world of scientific visualization to users of all skill levels. ezVIZ will initially be available in a batch version with a Web version due within the next year. Users can select from commonly run scripts or utilize their own to generate images in a number of standard file formats. ezVIZ will create images from their data while it still resides on the supercomputer. Once the images are finished, they can be easily moved to the user's computer.

The solutionHPC effort will be rounded out at a later date by ezSTOR, which will provide simple access to the mass storage facility. ezSTOR will utilize metadata stored in a database to help users track and cross reference past projects.

The overall solutionHPC concept was introduced in a recent edition of the *Resource*. With both ezHPC and ezVIZ preparing for beta tests, the ERDC MSRC felt this was a good time to offer users and developers a more in-depth view. The next two articles are written by the developers and provide an "under-the-hood" technical view inside both projects.



Technical Overview of ezHPC Architecture

By Weston Monceaux, Keith Rappold, and Patti Duett

In response to the need within the high performance computing (HPC) community for a user-friendly approach to resource access, the ERDC MSRC is bringing a project to completion that will provide users with an easy-to-navigate GUI. Beta testing will begin soon for this GUI called ezHPC, which will provide a robust, full-service access point to HPC assets.

ezHPC is an application that is easy to learn and can be accessed from anywhere using a standard Web browser or through a .Net client that can be downloaded. It provides the ability to authenticate users against the HPC Kerberos system, move files to and from the HPC systems, submit and monitor batch jobs, and access the mass storage facility (MSF).

This article presents a detailed inspection of the ezHPC architecture. For each component of the system, its interface structure, usage, and security implications are described. In turn, the HPC-issued Kerberos user utilities, the Web server components, and the database component are discussed.

Client Implementation

The ezHPC Client makes use of an application programming interface (API) Web service. Two client implementations are used. The first implementation client is a Windows-based application. The second implementation is a Web-based client. Both clients are designed for ease-of-use for common user-oriented actions on the HPC systems. The client performs all of its interaction with the HPC systems via the API Web service. The client is, therefore, considered a reasonably thin client. The bulk of the client processing involves accessing local PC resources (files, etc.). Otherwise, the API Web service does most of the work. The user, of course, will need network access and the ability to connect to the API Web service. The client needs only to have network access to the API Web service system and not any of the HPC systems, as all HPC system interaction is performed by the API Web service on behalf of the user.

Using the client, users are able to log in using Kerberos, move files to and from HPC systems, submit and

Figure 1. ezHPC user job status page

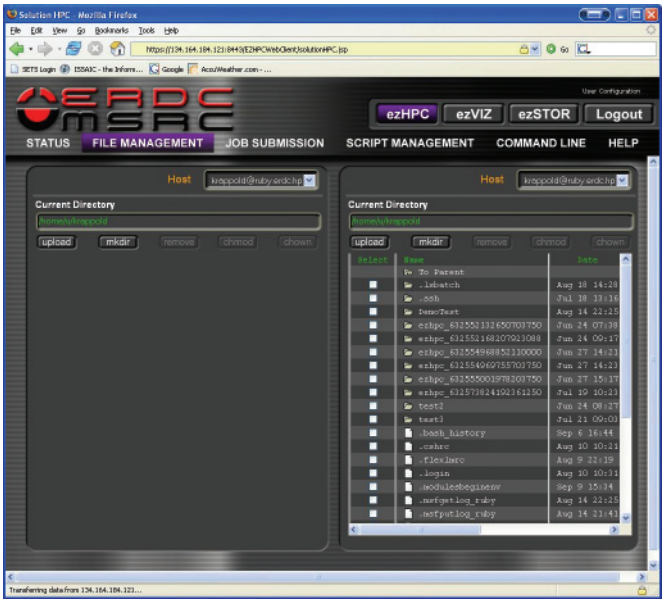


Figure 2. ezHPC file management page

Patti Duett
Chief, Software Engineering and Evaluation Branch
ERDC ITL

Weston Monceaux
Computer Scientist
ERDC ITL

Keith Rappold
Senior Software Systems Engineer
ERDC ITL

monitor jobs, access the MSF, create and save commonly used batch scripts, and view job history statistics. Figures 1 and 2 illustrate what users will see while checking job status and managing files.

Kerberos Utilities

The Kerberos utilities used are the Linux binary versions of the HPC-issued Kerberos and SSH utilities. Modifications have not been made to the tools. They are used "as is." The primary benefit of using the standard Kerberos tools is that if the tools are updated, few, if any, coding changes are necessary to utilize new versions of the tools.

The Web server is used to invoke a number of wrapper scripts, which in turn call the various Kerberos tools. These are Bourne shell scripts that take a number of parameters necessary to set up the proper environment variables to point to the current user's Kerberos ticket cache. The command to the HPC system can then be executed on behalf of the user. An Expect script is used for calling the `kinit` command.

Aside from ease of upgrading tool chains, the choice of using the standard Kerberos tools means that new security issues are not introduced by using nonstandard Kerberos utilities.

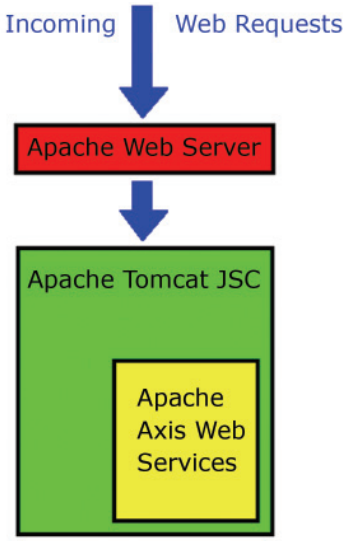
Web Server Components

This system is designed to be Web-based. All communication between a client and the HPC systems happens via the Web server components. Three software components make up the Web server components: the Apache Web server, Apache Tomcat Java Servlet Container, and the Apache Axis Web Service library.

The Apache Web Server (Apache) receives all incoming Web requests. The Apache Tomcat Java Servlet Container (Tomcat) runs as a separate process that is not accessible directly by the network. Apache forwards all requests to Tomcat. For those Web requests destined for the API Web service, Tomcat invokes the proper code within the Apache Axis Web Services (Axis) Web application. Axis runs as code within the Tomcat process.

The Web requests destined for the API Web service are SOAP requests, XML messages transported over HTTP. A SOAP request first arrives at Apache, which is then forwarded to Tomcat, which processes it into appropriate calls to the Axis code, which in turn invokes

The Web server components are organized as seen in the following diagram.



the API Web service code. Depending on the contents of the SOAP request, different actions are taken by the API Web service. Most actions involve invoking the Kerberos wrapper scripts to execute a command on an HPC system. The results of the SOAP request are returned back up the chain to the user as a SOAP response.

This structure is a Web-based remote procedure call (RPC) implementation. This is the basis for how all commands are issued to the API.

Apache, Tomcat, and Axis will remain dormant until something invokes methods from the API Web service. Methods are invoked by SOAP-aware applications. SOAP is a programming language neutral protocol. Several language libraries exist for it, including Microsoft .NET (C#, VB, etc.), Java, Python, PHP, Perl, and numerous others.

Tomcat is actually able to run as a stand-alone Web server. Apache is layered on top because it provides better control over SSL parameters. Apache is used to enforce AES or Triple DES encrypted SSL connections to the Web server. The clients connect to Apache using AES or 3DES SSL, and the requests are then forwarded (locally, not over the network) to Tomcat unencrypted.

The initial connection to the Web server is a presentation of authentication tokens that are used to try to authenticate the user. If `kinit` fails, an HTTP level

error is reported to the client. If `kinit` succeeds, subsequent requests are honored and passed along to the API Web service. The generated ticket granting ticket is passed back to the client to be used in future requests. The ticket is not stored on the Web server system. If a client submits a ticket as part of the HTTP authentication, it is used to process the request. All calls to the API Web service are logged.

Database

The database used is MySQL. It runs on the Linux system along with the Web server components. It is not accessible over the network. It is used to hold configuration information for all "configured" HPC systems. It also holds nonsensitive user information, such as job run statistics, application preferences, etc.

The MySQL database is used to hold all configuration information for the API Web service. The API Web service provides a uniform method interface across multiple HPC systems by creating mappings from the general desired functionality of the API to the specific implementation of that functionality on each HPC system. For instance, an API call to retrieve a directory listing makes use of the `ls` command on most HPC systems. Conceivably, however, an HPC system exists that uses some other command to retrieve a directory listing, such as `dir`. The database is used to maintain the function-to-command mappings for each HPC system. Each HPC system must have similar configuration information placed into the database before the API Web service will know how to properly interact with it.

The other type of information stored in the database is the user-centric data. The database is used to hold statistics on all job runs submitted through the API Web service. The following data are stored in the database:

- User Principle.
- HPC System configuration needed for ezHPC operation (e.g., location of utility binaries such as `ls`, `unzip`, `gzip`, `bsub`).
- User job information (e.g., job submission time, job name, job status).
- User-defined application scripts (e.g., `bsub` scripts for running a specific model).

The database is configured to communicate only with the Web service. When an API Web service method is called, the database is consulted to determine the proper commands to execute based on the selected host.

The database is protected from direct client access. Network connections are only permitted from the Web server. Database connections are encrypted with ssl using stunnel. Arbitrary database queries are not permitted via the API Web service.

Conclusion

The ezHPC architecture is designed with flexibility, extensibility, and security in mind. The authors believe the current design fulfills these design goals. In addition, the architecture will be continuously improved to provide additional scalability and robustness.

Glossary of Terms

3DES – Triple DES (also 3DES or DESede): a block cipher formed from the Data Encryption Standard (DES) cipher.

AES – Advanced Encryption Standard: a block cipher adopted as an encryption standard by the U.S. Government.

Apache Axis – an open-source, Java and XML-based Web service framework consisting of an implementation of the SOAP server and various utilities and APIs for generating and deploying Web service applications.

Apache Tomcat Java Servlet Container – provides an environment for Java code to run in cooperation with a Web server.

Bourne Shell – the default Unix shell of Unix Version 7.

MySQL – is a multithreaded, multiuser, SQL (Structured Query Language) Database Management System (DBMS).

PDA – Personal Digital Assistant: handheld devices that were originally designed as personal organizers, but became much more versatile over the years.

SOAP – Simple Object Access Protocol: a standard for exchanging XML-based messages over a computer network, normally using HTTP.

XML – Extensible Markup Language: a general-purpose markup language for creating special-purpose markup languages. Its primary purpose is to facilitate the sharing of data across different systems, particularly systems connected via the Internet.

ezVIZ – An Open-Source, Cross-Platform Visualization Tool

By Randall Hand and Paul Adams

This article will take a closer look at what makes ezVIZ an easy-to-use and powerful visualization tool for the HPC user community. Alpha testing has begun with ezVIZ, and some success has already been experienced. Two of these successes are discussed as case studies at the end of the article. The ERDC MSRC will start with the basics and build.

What is ezVIZ?

In 2000, the top computer in the world, IBM's ASCI White, had a maximal achieved performance of 2.38 TFLOPS. In 2004, 77 computers were more powerful than ASCI White, which is now 30 times slower than the fastest computer. This trend is expected to continue, with the first PFLOPS machine being available in 2009. With the ubiquitous availability of TFLOPS machines, the associated data are growing at an increasing rate. Terabyte data sets are now common.

Gaining useful information from such large data is a challenging task. This task often falls to the researcher or a visualization scientist. Extracting visual insight from a multiterabyte data set presents the researcher with several problems. These problems include transfer and storage of the data, graphics hardware to visualize it, as well as having visualization software capable of handling the data.

ezVIZ tackles the visualization portion of these problems by providing one of two mechanisms. The first, which is soon to be beta tested, is a batch visualization capability. This batch capability allows the users to create images from their data while it still resides on the supercomputer. These images, which are less than a few megabytes in size, can then be moved with ease to the researcher's workstation. Storage and network bandwidth are no longer a concern when visualizing the data. The second mechanism is to provide a Web interface to visualizing the data. This mechanism is currently under development.

ezVIZ Design Considerations

Several challenges, some of which are outlined above, were the driving force behind the design of ezVIZ. The first of these challenges is simply the data itself. This includes managing data locality (i.e., the transfer and storage of the data). By leaving the data on the supercomputer, the researcher does not need to worry, at least for ezVIZ, about moving data from one machine to another. In addition, this includes the format of the data itself. Since a standardized format does not exist for writing data from simulations run on the supercomputers, different researchers use different formats. The desire is for ezVIZ to be able to read in a variety of formats.

The second challenge is to have the necessary software tool to visualize the data. ezVIZ could be written from "scratch" to meet this design point. However, reinventing the wheel was undesirable. Several commercial off-the-shelf (COTS) products were examined, including Ensight and VTK. The difficulty with using a product like Ensight, which supports batch visualization, is that the source code is unavailable. This limits what a researcher can accomplish with the code. In addition, Ensight is not available on all platforms. VTK was chosen because it is an open-source visualization toolkit. Since VTK is open-source, it can be ported to new machines, as they become available and extended to meet the specialized needs of HPC users.

Another constraint is that most visualization is performed on machines that have graphics hardware. Most supercomputers, with the exception of some from SGI, do not have graphics hardware available on them. The Mesa three-dimensional (3-D) graphics library is an open-source graphics library with an application programming interface (API) that is similar to OpenGL. Mesa allows for off-screen rendering by using the central processing unit (CPU). It is not dependant on any graphics hardware. In addition, Mesa support is included in VTK. Mesa implements the latest version of OpenGL, which is currently Version 1.5.

Randall E. Hand
Visualization Scientist
ERDC MSRC

Paul Adams
Scientific Visualization Lead
ERDC MSRC

Again, since Mesa is open-source, it can be ported to new machines, as they become available, and compiled on them.

Usage

ezVIZ supports several modes of operation:

1. A C/C++ API for directly linking into simulations, or writing one's own small visualization applications.
2. A stand-alone application that reads configuration files that can be run via "shell" or "exec" commands within the simulations.
3. A stand-alone application that reads configuration files that can be run on previously generated data.

Depending on the desired complexity of the scene, the code or configuration files can be as little as five lines. Also, ezVIZ does not require that individual scenes be constructed for multiple renderings. A single scene can be configured once and passed to several rendering functions to generate multiple outputs or can be reconfigured between rendering functions to generate slightly varying outputs like rotations.

The stand-alone application supports all the features of the API, but in a complete bundled package. Using this application or separate visualization programs of only a few dozen lines of code and compiling them separately enables the execution through various shell or exec functions used by the simulation. This adds a level of error recovery, by separating the visualization from the simulation, such that any errors or crashes in the visualization will not affect the running simulation. This capability also allows for postprocessing of data after the simulation has completed.

The first step of using ezVIZ is inputting data. ezVIZ requires that the input data be written to a disk, so that they are in a standard and easily accessible format. Data that resides solely in memory are currently not compatible with ezVIZ. Since ezVIZ is built on top of VTK, it supports all of the formats that VTK supports, and a few more added outside of VTK, as can be seen in Figure 1.

Various rendering methods require different types of input data. Where possible, data conversion is done automatically; otherwise, ezVIZ simply displays a warning message and fails to generate the desired output.

PNG, BMP, JPG, TIF, PNM	Classic Image formats
VTK	VTK Legacy Format
OBJ	WaveFront OBJ Files
PLY	Stanford University Format
POLY	Side Effects PRISM ASCII Format
STL	Common Stereolithography Format
BYU	Brigham Young University Format
CASE, SOS	Ensight
DEM	Digital Elevation Maps
XYZ	Plot3D
UG	Unigraphics Facet Format
MHA, MHD	UNC MetaImage Format
FACET	Facet Files
CUBE	Gaussian Cube Files
INP	AVS "UCD" Format
VTP, VTU, VTR, VTS, VTI	VTK XML Formats
XDMF	Extensible Data Model Format, by ARL
XMDF	Extensible Data Model Format, by ERDC/BYU
PDB	PDB Molecular Data Files

Figure 1. Supported input formats

The result of an ezVIZ operation is more files, most commonly images. Several image formats are supported, and at many varying sizes. Several common formats are supported including PNG, JPG, TIF, BMP, and PNM. These images can be generated at arbitrary sizes, even exceeding OpenGL and Mesa's 4096×4096 texture limitations, and can be titled and time-stamped. Bounding boxes and color bars can also be enabled for further information. All of these functions are user-customizable and can be turned off if desired.

Certain rendering techniques (e.g., isosurfacing) generate or extract geometry that is useful for other functions, such as third-party rendering packages. ezVIZ supports saving the geometry when using these techniques as shown in Figure 2.

OBJ	WaveFront OBJ – Stores Geometry Only
VRML	Virtual Reality Modeling Language – Stores Geometry and Color Information
PLY	Stanford University PLY Format – Stores Geometry and Color Information
STL	Stereolithography Format – Stores Geometry Only

Figure 2. Supported output polygonal formats

In addition to simply storing these output formats, ezVIZ supports a basic set of postprocessing tools for geometric outputs:

- Cleaning – removing duplicate vertices and edges.
- Decimation – Quadric Decimate to reduce the size and complexity of the mesh.
- Triangulation – converting a polygonal mesh to a triangular mesh (required for some output formats).

Using the polygonal output mode of ezVIZ, cleaning and decimation of surfaces or models read from other analysis or visualization packages can be done in a batch mode.

ezVIZ is capable of visualizing data in several ways. These include rendering a polygonal data set (such as a model), extracting a contour or isosurface, using flow glyphs, streamlines, and volume rendering.

The basic model-rendering mode loads a simple polygonal data set from a disk, such as an isosurface or bounding surface generated by another program or another ezVIZ rendering process. Once loaded, it can simply be rendered to an image or saved as another model to a disk. The latter mode is useful when combined with the Cleaning and Decimation tools to reduce the complexity of models.

The contour extraction technique works with grid data sets and extracts isocontours or isosurfaces. Grid data sets can be either 3-D grids or 2-D grids (where the third dimension is collapsed to a plane). The resulting isosurfaces can be colored by any additional field and then saved as an image or a model. The coloring operation can be mapped to an additional scalar or any component or magnitude of a vector field. If the input data set uses cell data instead of point data, then the data will automatically be converted into the appropriate format.

When using flow glyphs, a grid data set is loaded (as stated before, either 2- or 3-D), and one of many simple models, ranging from a 3-D arrow to a simple vertex point, is placed at each data point. The resulting complexity can be overwhelming to many systems, so a masking function is in place to allow a statistically random or uniform scattering of points to be used in place of the entire data set. The models can be oriented to a vector field, or left unoriented, and can be scaled by a scalar field in the data set. They can also be colored by any other field or the resulting vector magnitude or scaling factor, and then written to a disk as an image or model.

Streamline generation loads a grid data set (2- or 3-D) and shows streamlines passing through a specified point. More than one streamline can be generated at a time using the spherical or line sources. The resulting lines can be colored by additional fields in the data set or by several parameters resulting from the streamlines calculation (normals, rotation, vorticity, etc.). Several parameters exist to control the types of calculations used for the placement of the streamline and how it will terminate. The result of a streamline rendering can only be saved as an image.

Finally, volume rendering takes a grid data set and then generates an image using ray-casting techniques. One is able to map opacity and color to separate values or to the same value. The result can only be saved as an image.

Conclusions

ezVIZ is one portion of a larger solution called solutionHPC, to enable high performance computing users to easily generate and understand their data. By basing ezVIZ on open-source visualization code, cross-platform migration is achieved. HPC users can either integrate ezVIZ into their own code or run it separately. ezVIZ is currently undergoing testing with a select group of users at the ERDC MSRC.

Acknowledgments

Thanks to Doug Dommermuth for the Breaking Waves data sets. Thanks to Randy Moses for the Backhoe SAR data set.

Case Studies

Case Study No. 1: Volume-rendered backhoe

Problem: A user has a data set from a simulated synthetic aperture radar image of a backhoe from which he would like to extract a volume-rendered image. The size of the data set is 1024^3 and is in the Visualization Toolkit (VTK) legacy format. The resulting 1280×960 image should be rendered so that it masks any noise in the data while clearly showing the backhoe in the center.

Solution: An ezVIZ program was written in C to generate the desired scene. Sixteen processors were used to ray cast the scene in parallel. The relevant code can be seen in Figure 1. The image, which can be seen in Figure 2, took 13.3 minutes to generate on an Onyx 340 with 600-MHz MIPS processors.

The same data set was processed again, but the resulting image was 6400×3600. This requires tiling of the image, as it exceeds the 4096^2 texture limit in OpenGL. Sixteen processors were used to ray cast in parallel, and 360 images were generated from various angles to create a full rotation. Each frame took an average of 30 minutes, with the entire process taking approximately 6.5 days working completely unattended on an Onyx 340.

```
// 1-time setup
scene = ezVISInitialize();
strcpy(scene->input_filename, argv[1]);
scene->output_height = 3600;
scene->output_width = 6400;
scene->showTitle = 0;
scene->showTimeStamp = 0;

scene->enableColorMap = 1;
scene->showColorBar = 1;
sprintf(scene->colorField, "volume_scalars");
scene->colorBarOrientation = ezVIS_OrientHorizontal;
scene->colorBarLocation = ezVIS_SideBottom;
scene->colorFieldComponent = ezVIS_RangeX;
scene->boundingBox = 0;

scene->Elevation = -75;
scene->Pitch = 1;
scene->Zoom = 5;

// Build the colormap, since the default uses full opacity at all values
ezVIS_AddColorMapEntryRGB(scene,  0.0, 0,0,0,0.00);
ezVIS_AddColorMapEntryRGB(scene, 36.5, 0,0,1,0.16);
ezVIS_AddColorMapEntryRGB(scene, 73.0, 0,1,1,0.33);
ezVIS_AddColorMapEntryRGB(scene,109.5, 0,1,0,0.50);
ezVIS_AddColorMapEntryRGB(scene,146.0, 1,1,0,0.66);
ezVIS_AddColorMapEntryRGB(scene,182.5, 1,0,0,0.83);
ezVIS_AddColorMapEntryRGB(scene,219.0, 1,0,1,1.00);
ezVIS_AddColorMapEntryRGB(scene,255.0, 1,1,1,1.00);

ezVISSetNumberOfThreads(24);
for(scene->Roll = 0; scene->Roll < 360; scene->Roll++) {
      sprintf(scene->output_filename, "frame%04i.png", scene->Roll);
      ezVISRenderVolume(scene);
}
```

Figure 1. Source for the backhoe volume rendering

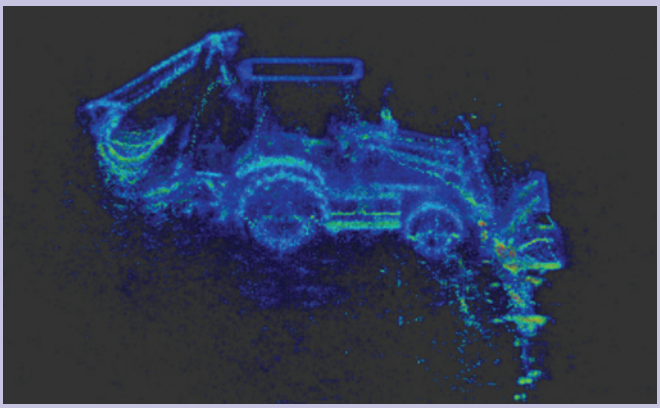


Figure 2. Small backhoe volume rendering

Case Study No. 2: Isosurface of a ship's breaking waves

Problem: A user has a large time-varying data set of a ship's wake in water, from which he would like to generate a movie. The data set is 600 time-steps of a 512×128×128 grid containing a scalar (fraction of water in a given cell) and a vector (velocity) at every point. The resulting movie should contain 600 frames showing an isosurface of the fraction field, which indicates the water surface, and should be color-mapped according to flow-velocity magnitude.

Solution: A small ezVIZ script, as opposed to the C program used above, was created to generate the requested image using the out-of-core tool. All 600 frames were processed through ezVIZ, writing to a disk as 800×600 PNG image. The script to process one file can be seen in Figure 3. One of the resulting images can be seen in Figure 4. The script was wrapped in a shell script to process multiple files simultaneously and automatically. Twenty-four frames were processed at a time, each frame taking approximately 15 to 20 seconds. The entire process took approximately 7.5 minutes.

The same data set was processed again, but written as 6400×3600 PNG images. This requires tiling of the image, as it exceeds the 4096^2 texture limit in OpenGL. The resulting images took ~2.5 minutes each, and the entire process took approximately an hour.

```
[OUTPUT]
Width=800
Height=600
File=time-0.png
Format=png

[INPUT]
File=time-0.vtk


[RENDER]
Operation=contour
Scalarfield=Fraction
Component=scalar
Isovalue=0.5
Elevation=-30

[COLORMAP]
Colorbar=bottom
Stretch=off
Colorfield=Velocity
Colorcomponent=magnitude
Colormapentry1=0.00=HSV(0.6667,1,1)
Colormapentry2=1.50=HSV(0.3333,1,1)
Colormapentry3=3.00=HSV(0.0000,1,1)
```

Figure 3. Script for the breaking waves isosurface

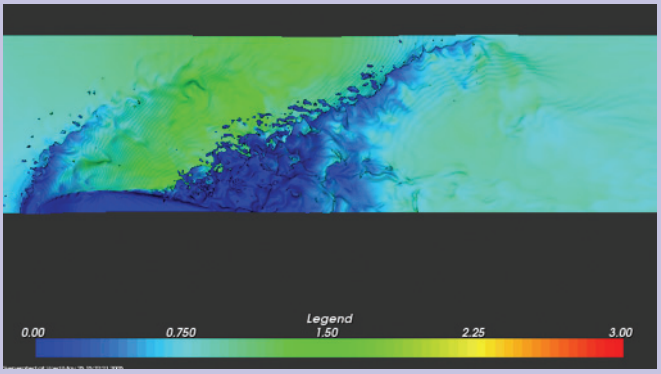


Figure 4. Small breaking waves frame

About PET

By Dr. Jim Pennington

The Programming Environment and Training (PET) program works in the DoD's High Performance Computing Modernization Program (HPCMP) to provide expertise, technical support for computing resources, education, and training. This program is in the fifth year under the current contract with the HPCMP. PET is a national team of academic institutions and industry partners that work side-by-side with DoD researchers in several computational technology areas (CTAs). The CTAs are managed by assigning the technical areas to one of four Components. A Component is hosted by one of the four Major Shared Resource Centers (MSRCs): the Naval Oceanographic Office (NAVO) at Stennis Space Center, Mississippi; the Aeronautical Systems Center (ASC) at Wright-Patterson Air Force Base at Dayton, Ohio; the Army Research Laboratory (ARL) at Aberdeen Proving Grounds, MD; and the Engineer Research and Development Center (ERDC) at Vicksburg, MS.

The ERDC MSRC is host to PET Component 3, which provides support to the DoD HPC user community in the Computational Fluid Dynamics (CFD) and Computational Structural Mechanics (CSM) CTAs. Component 3 is also responsible for coordinating training and outreach at DoD HPC sites through the crosscutting functional area of Education, Outreach, and Training Coordination (EOTC).

ERDC MSRC PET summer interns

Component 3 onsite resource persons at the ERDC MSRC are Dr. Nathan Prewitt for CFD and Dr. Richard Weed for CSM, both of whom are employed by Mississippi State University. Also onsite for Environmental Quality Modeling and Simulation (EQM) is University of Texas employee Dr. Jeff Hensley. The PET team at the ERDC MSRC is available to assist all HPC users in DoD. Assistance in the other CTAs is available

Dr. Jim Pennington is the new PET Component 3 Point-of-Contact. This component, hosted by the ERDC MSRC, includes the CFD and CSM CTAs and the responsibility for coordinating training and outreach at DoD HPC sites through the crosscutting functional area of EOTC.

Dr. Pennington retired March 3, 2005, as Director of the Graduate Institute, an extended campus located at ERDC, Vicksburg, formed to offer advanced degrees from its member universities, Mississippi State, Louisiana State, and Texas A&M. He was also the Program Manager of the ERDC Science and Technology Laboratory Personnel Demonstration Project, as well as serving ERDC in many other capacities.

Dr. Pennington earned his BS in zoology and MS in biology from Southeastern Louisiana University, Hammond, and his PhD in Wildlife and Fisheries Sciences from Texas A&M University. He is a member of the American Fisheries Society, the Mississippi Chapter of the American Fisheries Society, and the American Society for Engineering Education. He serves as an adjunct professor for the Departments of Biology at Southeastern Louisiana University and University of Southern Mississippi.

from PET university faculty and onsites leads at other MSRCs and DoD facilities.

The Component 3 goal in CFD and CSM is to improve DoD's simulation capability. The CFD team provides support in the development and use of flow solvers for challenging CFD problems involving moving body dynamics and complex turbulent flow. The CSM team provides support to the CSM user community in algorithm development for improved shock physics and penetration mechanics simulations. Through these efforts, the CFD and CSM teams improve the capability of DoD users to efficiently perform CFD and CSM simulations in the HPC environment. The goals in EOTC are to coordinate all training offered under PET, to interface with educational institutions, to assist in addressing the shortage of skilled employees by tapping the pool of underrepresented groups in science and technology, and to provide leadership in outreach activities with Minority Serving Institutions (MSIs).

MSI Activities

The PET MSI outreach activities in 2005 included MSI participation in HPC projects; involving 46 students in the 2-week-long Summer Institutes that were held at the University of Hawaii, Central State University, Florida Institute of Technology, and Jackson State University (JSU); and mentoring students with work assignments in the 10-week-long HPC Summer Intern programs that are hosted by the four MSRCs. The ERDC MSRC hosted 6 of the 26 summer interns during May, June, and July. These interns represented a variety of backgrounds, interests, and goals for their future. With the rich variety of technical work at ERDC, the MSRC was able to offer the students a variety of intern projects so that each would benefit from experiences by working side-by-side with their mentors.

JSU HPC Summer Institute

The JSU HPC Summer Institute was held July 11-12, 2005, at JSU's MS e-Center under the direction of Drs. Robert Whalin and Shahrouz Aliabadi. Twelve JSU students were selected to participate in the program. Activities for the students during the 2 weeks included workshops, projects, seminars, and lectures covering the following topics: high performance computing and applications, programming tools, numerical methods and applications, parallel programming, and mesh generation. The students were also treated to several invited presentations – two by ERDC researchers Drs. Jane Smith and Charlie Berger and one by Charles Nietubicz, ARL MSRC Director.

The JSU students also spent a day touring the ERDC laboratories in Vicksburg where they saw firsthand the application of the concepts covered during their lectures. While at ERDC, the JSU students also had the opportunity to interact and share experiences with the ERDC MSRC summer interns.

Student feedback included the following comments:

"The information that I have I can apply to biology and other sciences. I think that is awesome because I had the opportunity to view things outside the biology lab."

"The lectures were very informative, but the hands-on experience was a huge plus. We were physically involved with everything from actually writing FORTRAN programs to generating meshes and actually implementing various formulas to run real-time simulations."

"This has been a good experience for me because it gave a wide spectrum of occupations in the engineering field outside of those that are generally more well-known. It widened the outlook I had on my future career and opened me up to other possible choices and opportunities. The valuable knowledge I gained will definitely help me in my future endeavors."

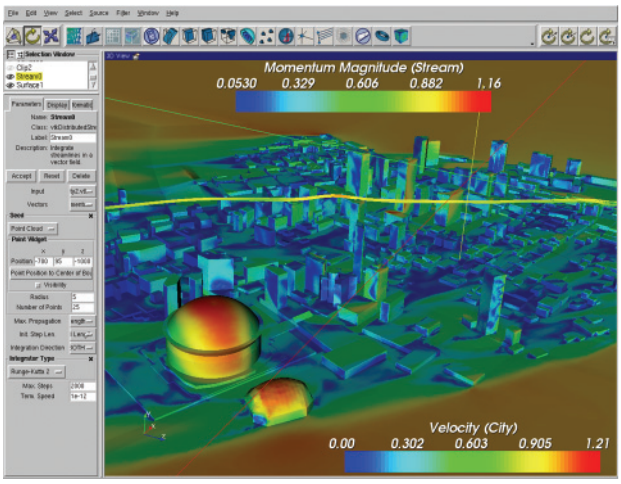
How to Find PET Information

Information about the PET program can be found on the Online Knowledge Center (OKC) at https://okc.erdc.hpc.mil. The OKC resides at the ERDC MSRC but provides information on all components of PET. Users can find information about training, projects, reports, presentations, and the descriptions and contact information for all technical areas supported under the PET program.

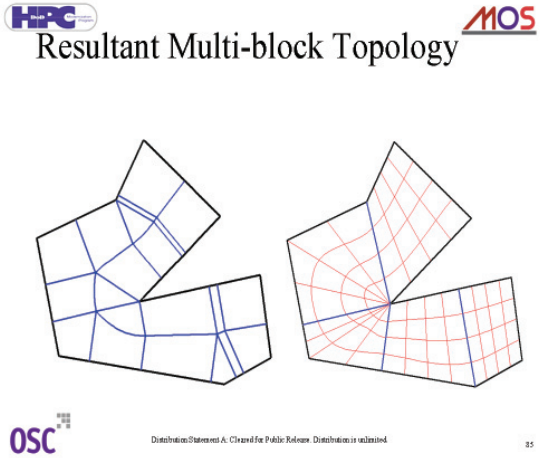
JSU students who participated in the JSU HPC Summer Institute

ERDC MSRC PET Summer Interns . . . . . .

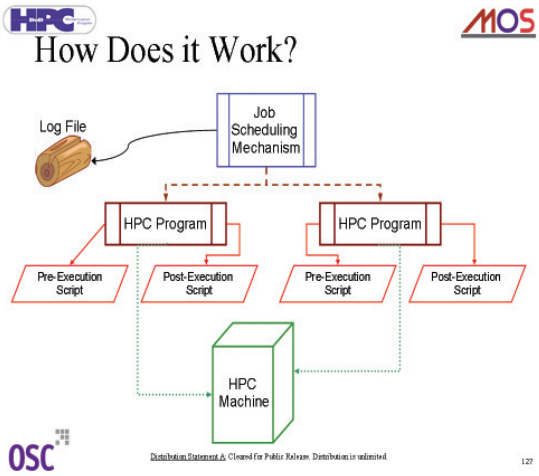
Veronica Cornner is a graduate student at the University of Alabama in Birmingham majoring in mechanical engineering. She received her undergraduate degree in physics from Fisk University. While at Fisk, she was captain of the track and cross-country teams, and she would some day like to run competitively. Last summer, Veronica participated in the Summer Intern Program at Wright Patterson Air Force Base. Paul Adams and Randall Hand served as her mentors this summer, and her project involved the visualization of a wind field through New Orleans. Her project has application to emergency planners who are interested in visualizing the flow of contaminants, such as chlorine gas escaping from a damaged barge, through the city.

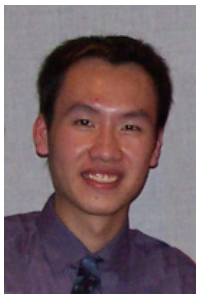


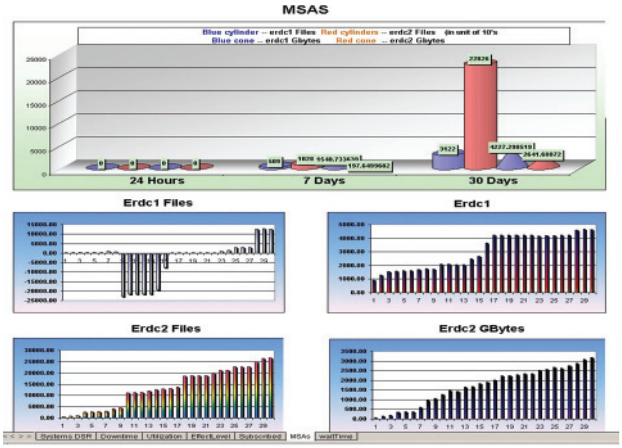
Si Loi Leung is a senior at the University of Hawaii majoring in mechanical engineering. His interests, other than computing and programming, include tennis, bicycling, going to the beach, and video games. He was an intern at the ERDC MSRC last summer and "loved it." Si Loi was mentored by Nathan Prewitt, and his project involved automatic structured grid generation in two dimensions based on the use of medial axis graph. His work will be used in generating grids for the calculation on contaminant transport through a building.



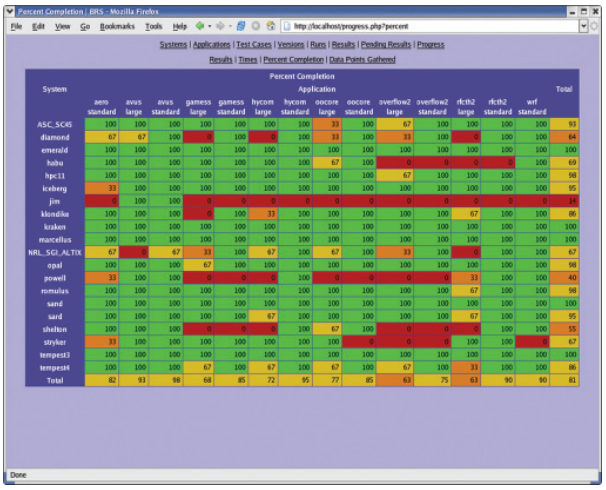
Corey Bordelon is a senior at the University of Louisiana, Monroe, majoring in computer science. This is his second summer to participate in the Intern Program, and he worked under the direction of Dr. Jeff Hensley. Corey developed a framework for automated sensitivity analysis that makes it easy for EQM users to submit and monitor large numbers of model runs on the HPC platforms. This is useful, for example, for parameter estimation when attempting to determine physical properties of soil types in groundwater flow. In his spare time, Corey enjoys hunting, fishing, working with the mini-zoo he calls home, learning new programming languages, Web development, and cooking.



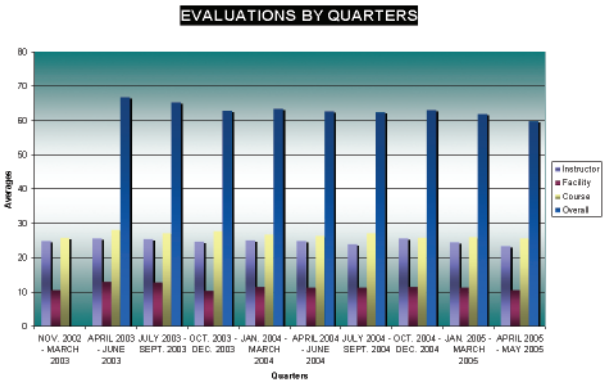




Jian Zhong "Dan" Li *is* beginning his junior year at Bucknell University in the Computer Science in Engineering Program. This is the second year that he has participated in the ERDC MSRC PET Summer Intern Program under the direction of his mentor, Jay Cliburn. Dan's summer project, an automated graphical report that shows daily performance measurements for each of the MSRC's HPC machines and mass storage archive systems, was extremely useful to his mentor. Dan is involved in a number of student organizations at Bucknell, including the Chinese Culture Association, Student Asian Awareness at Bucknell, Vietnamese Student Association, Japanese Club, Association for Computing Machinery, and the Shotokan Karate Club.



Matthew Bennett is a senior at Mississippi State University majoring in computer science. His experience includes working with the Information System Division of Wal-Mart, Inc., and he has also been gaining experience in Human and Systems Engineering through the Center for Advanced Vehicular Systems at Mississippi State. His mentor this summer was Dr. Alvaro Fernandez, and his project involved the development of a user-friendly Web application to facilitate the submission and retrieval of Technology Insertion benchmark data to provide a scientific basis for comparison of HPC systems.



Tanjala Purnell is a recent graduate of Tougaloo College where she earned her BS degree in computer science with summa cum laude honors. Her hobbies include listening to good music, reading, playing musical instruments, and just having a nice time with "cool" people. She will begin graduate studies at Ohio State University this fall semester. Tangala's mentors were Reggie Liddell and John Mason, who directed her project, which included the analysis of Oracle-based PET training evaluation data. The information gathered by Tangala will be useful for decision making about how resources can be used to further enhance the quality of PET training while maintaining focus on DoD user requirements.

Storage Monitoring and Forecasting at the ERDC MSRC

By Teresa Moncrief and David Sanders

The ERDC MSRC has initiated a program for monitoring storage utilization by tracking storage usage for each subproject allocated computing hours by the DoD High Performance Computing Modernization Program (HPCMP). HPCMP users will store data in subproject specific directories on the data management system (DMS) and categorize data currently in their archive home directory. While this change may cause some short-term inconveniences, monitoring storage utilization is needed to do a better job of forecasting storage requirements to prevent service disruptions.

Impetus for Change

This project is the fruit of an unexpected storage capacity shortage experienced in FY 2005. At the beginning of FY 2005 (October 2004), the ERDC MSRC projected the DMS usage to be within current capacity through the end of the FY. At the time, the DMS had 140 TBytes of available tape storage with an additional 160 TBytes to be added in April 2005. New data being written averaged approximately 8 TBytes a month. By December 2004, the forecast changed with the abrupt increase of data storage by 23 TBytes per month during November and December. Projections suggested that file storage space would be exhausted by March 2005.

In January 2005, the ERDC MSRC contacted users with proportionally larger amounts of data on the DMS and requested that they reduce their file storage. By late February, consenting users had freed about 50 TBytes of space, a 15-percent reduction in overall storage. That, combined with the delivery of new tapes and additional capacity that was realized once the new Disaster Recovery (DR) site was fully populated, brought the Center out of the immediate crisis.

Analysis of Available Data

Once the capacity shortage had been remedied, the ERDC MSRC began researching the available data to determine if the sudden surge in storage utilization had been predictable. Analysis revealed that the drastic increase in data storage rates in FY 2005 was largely the result of users associated with a small number of subprojects. Based on the data available locally and through the Information Environment (IE) database, the ERDC MSRC realized that it could not have anticipated the surge of the subprojects in storage utilization. Even now, the ERDC MSRC does not know whether this increase was an anomaly or a vision of the future.

Teresa Moncrief, Storage and Monitoring Forecasting Lead, and David Sanders, HPC Systems Specialist, both of the ERDC MSRC

The Center needs a mechanism for anticipating the storage needs of the user community so that the right amount of storage is available at the right time.

The biggest limitation for forecasting is the absence of qualitative data about the files stored on the DMS. To date, the DMS has been seen as an extension of the HPC systems. It has simply been a remote file system on which to store user data. Because data are only associated with users and users are often on multiple subprojects, determining what subprojects or codes or computational technology areas (CTAs) have greater storage requirements is difficult. By associating the files stored on the DMS with subprojects, the ability to perform this type of analysis becomes possible.

Storing Files in FY 2006

Beginning in FY 2006, all files sent to the DMS will be placed in a subdirectory specifically designated for the subproject under which the files were created.

These new subdirectories have already been created under each user's top-level archive directory, referenced by the environment variable **$ARCHIVE_HOME**. The names of the subdirectories match the names of their corresponding subprojects. For example, if a user is eligible to run jobs at ERDC under subprojects *abcde111* and *fghij222* on one system and *klmno333* on a second system, then subdirectories named *abcde111*, *fghij222,* and *klmno333* will be in the user's archive directory on the DMS.

Users are encouraged to begin making changes to scripts and procedures to begin storing all new archival

data in the subproject subdirectories. For example, if a job is to be run under subproject *pqrst444* and needs to store its output data on the DMS, the storage commands within the job would need to change as in the following script fragments:

Old Commands:

```
$ archive mkdir -C $ARCHIVE_HOME/run37
$ archive put -C $ARCHIVE_HOME/run37 datafile
```

New Commands:

```
$ archive mkdir -C $ARCHIVE_HOME/pqrst444/run37
$ archive put -C $ARCHIVE_HOME/pqrst444/run37 datafile
```

In addition, all files and directories currently on the DMS must be relocated into an appropriate subproject subdirectory. To accomplish this relocation, users can directly log on to the DMS and move the files or use the ERDC Mass Storage utilities. To directly log on to the DMS the command `getarchost` may be used on any ERDC system to display the address of the DMS server on which the user's archive directory is located. For example,

```
$ getarchost
silver.erdc.hpc.mil
```

Relocation of existing files

The new storage scheme will require that users categorize existing files into the available subproject subdirectories. The exceptions to the relocation requirement are any personal login or shell-history dot-files such as .login, .cshrc, .profile, .history, etc. These files should remain in the top-level **$ARCHIVE_HOME** directory.

Relocation is done using the Unix `mv` command to relocate files instead of using the `cp` command to copy them. `mv` works almost instantaneously because it simply changes a file pointer. `cp` must first unmigrate the data from tape to disk before it then makes a second copy of the file in the new location, thus consuming double the amount of space.

In 2006, permissions for all top-level archive directories will change to read-only mode to prevent new files from being stored into that location. From that time, files can only be stored under subproject subdirectories.

Users may contact the ERDC MSRC HPC Service Center with questions or to request assistance in modifying job scripts and moving files.

DoD Supercomputing Center at ERDC to be Featured Worldwide Once Again

By Rose J. Dykes

Soldiers Radio and Television (SRTV) recently returned to ERDC to film various ERDC Vicksburg research projects. The DoD High Performance Computing Center was among the groups filmed at ERDC by SRTV reporter Jini Ryan.

SRTV, the Pentagon Channel, the Armed Forces Network, and over 900 cable access channels will air a 30-minute ERDC news show in the near future including supercomputing and scientific visualization facilities located in the ERDC Information Technology Laboratory (ITL). SRTV filmed these facilities once before in 2002.

In his on-camera interview with SRTV, Dr. Jeffery Holland, ITL Director, focused on the capabilities of high performance computing, what it allows DoD researchers to do today that was not attempted a few years ago, and where the ERDC super-computing center ranks in the world. Dr. Michael Stephens, ERDC MSRC Scientific Visualization Center, explained the benefits of visualization of computing data in his on-camera interview, including significantly cutting defense system costs by shortening the design cycle and reducing reliance on expensive and destructive live experiments and prototype demonstrations.

Dr. Jeffery Holland (center), ERDC ITL Director; Wayne Stroupe (left), ERDC Public Affairs Office; and Jini Ryan, SRTV reporter, prepare for Dr. Holland's filming of his discussion on high performance computing

Dr. Michael Stephens, ERDC MSRC Scientific Visualization Center, is filmed by SRTV reporter Jini Ryan as he discusses benefits of scientific visualization

Helping Shape the Future

Job Shadowing

Job Shadowing is designed to give high school students an opportunity to "shadow" or follow a workplace mentor to gain firsthand experience and knowledge about a particular job. Paul Adams (right), ERDC MSRC Scientific Visualization Lead, is shown with Vicksburg High School student Jennifer Smith, who shadowed him.

Engineer Camp

David Stinson, ERDC MSRC Assistant Director, discusses high performance computing with high school students from Mississippi and Louisiana attending Engineer Camp, an annual activity sponsored by the Society of American Military Engineers for one week at ERDC.

Student Internships and Institutes

ERDC MSRC Director John E. West takes time to make a special presentation to the PET Interns and students attending the Jackson State University Institute.

ERDC MSRC Makes 11 Presentations at UGC 2005 and Brings Home Two Hero Awards

By Rose J. Dykes

The ERDC MSRC presented the most papers of any shared resource center attending the annual DoD HPCMP Users Group Conference (UGC). It also participated in the conference poster session. This year's UGC was held in Nashville, Tennessee, June 27-30, 2005. Approximately 400 attendees gathered for the conference, representing either the HPCMP computing centers or the users of the resources of the centers.

Cray Henry, the HPCMP Director, presented two ERDC MSRC team members with Hero Awards at the conference. One recipient, Dr. Ruth Cheng, received the Hero Award for the category *Up and Coming Within the HPCMP*, for making significant contributions in science or engineering. Receiving the Hero Award for the category *Innovative Management* was Greg Rottman, for demonstrating creative business practices to improve overall HPCMP business model.

Hero Award Recipients

Greg Rottman
Assistant Director
ERDC MSRC

Dr. Ruth Cheng
Computational Engineer
ERDC MSRC

ERDC MSRC Paper Presentations

These papers will be available soon on the IEEE Conference Proceedings Web site at

http://csdl2. computer.org/persagen/DLPublication.jsp?pubtype=p&acronym=DOD_UGC

Performance Comparisons of Visualization Architectures – *Paul Adams*

CCM-, CEA-, and CSM-Based HPCMP Systems Assessment Using GAMESS, OOCORE, and RFCTH-2 – *Dr. Paul Bennett*

Three-Dimensional Target Visualization from Wide-Angle IFSAR Data – *Tom Biddlecome*

CFD-Based HPCMP Systems Assessment Using AERO, AVUS, and OVERFLOW 2 – *Dr. Samuel Cable*

Parallel Algorithms for a Coupled System on Multispatial and Multitemporal Scales – *Dr. Ruth Cheng*

ezVIZ—An Open-Source, Cross-Platform Visualization Tool – *Randall Hand*

CWO-Based HPCMP Systems Assessment Using HYCOM and WRF – *Carrie Leach*

Implementation of Storage Allocations at the ERDC MSRC – *Teresa Moncrief*

ezHPC—Easy Access to High Performance Computing – *Kevin Rappold*

Role of Algorithms in Understanding Performance of the TI-05 Benchmark Suite – *Dr. Fred Tracy*

Other ERDC Paper Presentations

Comparison of Numerical and Experimental Results for Nine Building Airblast Experiments – *Dr. James Baylot*

Seismic Propagation from Humans in Open and Urban Terrain – *Dr. Stephen Ketcham*

ERDC MSRC Poster Presentation

Load Test: Benchmarking Throughput – *Dr. Paul Bennett*

(Left to right) John E. West, ERDC MSRC Director, with U.S. Senate Appropriations Committee, Roger Cockrell, staff member, Scott O'Malia, Majority Clerk, and Keith Kennedy, Majority Staff Director, August 9, 2005

BG William T. Grisoli (center), Commander Northwestern Division, Incoming Commander, North Atlantic Division, U.S. Army Corps of Engineers; Dr. William D. Martin (left), Deputy Director, ERDC Coastal and Hydraulics Laboratory; and Dr. Jeffery P. Holland (right), Director, ERDC Information Technology Laboratory; April 20, 2005

John E. West (right), and Stephen Coakley, Deputy Chief of Staff, Resource Management, U.S. Army Corps of Engineers, Washington, DC, April 11, 2005

BG Daniel J. Kaufman (second from right), Dean of Academic Board, U.S. Military Academy, West Point, New York, and Party with David Stinson (far right), Assistant Director, ERDC MSRC, February 18, 2005

Computational Technology Areas Advisory Panel, (seated left to right) Dr. Michael R. Macedonia, Forces Modeling and Simulation Lead; Dr. Ruth Pachter, Computational Chemistry and Materials Science Lead; Dr. Larry Davis, Deputy Director, HPCMP; and Dr. John M. Harding, Climate/Weather, Ocean Modeling and Simulation Lead; (standing left to right) Dr. Arunachalam M. Rajendran, Computational Structural Mechanics Lead; Dr. Barry S. Perlman, Electronics, Networking, and Systems/C41 Lead; Dr. Jeffery Holland, Environmental Quality Modeling and Simulation Lead; Dr. Robert L. Meakin, Computational Fluid Dynamics Lead; Dr. Anders Sullivan, Signal/Image Processing Lead; Jere J. Matty, Integrated Modeling and Test Environments Lead; and Dr. Kueichien C. Hill, Computational Electromagnetics and Acoustics Lead, February 17, 2005

52nd Engineer Battalion, Fort Carson, Colorado, February 10, 2005. In 1999, this battalion was established as one of the first multicomponent engineer battalions in the Army, consisting of Active Duty, Army Reserve, and National Guard companies, providing construction and combat engineer support

Below is a list of acronyms commonly used among the DoD HPC community. These acronyms are used throughout the articles in this newsletter.

AERO	Aeroelastic CFD/CSM
AES	Advanced Encryption Standard
API	Application Programming Interface
ARL	Army Research Laboratory
ASC	Aeronautical Systems Center
AVUS	Air Vehicles Unstructured Solver
CCM	Computational Chemistry and Materials Science
CEA	Computational Electromagnetics and Acoustics
CFD	Computational Fluid Dynamics
CHARTD	Computational Hydrodynamics and Radiative Thermal Diffusion
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CSM	Computational Structural Mechanics
CTA	Computational Technology Area
CTH	CSQ to the Three-Halves
CSQ	CHARTD Squared
CWO	Climate/Weather/Ocean Modeling and Simulation
DMS	Data Management System
DoD	Department of Defense
DR	Disaster Recovery
EOTC	Education, Outreach, and Training Coordination
EQM	Environmental Quality Modeling and Simulation
ERDC	U.S. Army Engineer Research and Development Center
GAMESS	General Atomic and Molecular Electronic Structure System
GUI	Graphical User Interface
HPC	High Performance Computing
HPCMP	High Performance Computing Modernization Program
HTTP	Hypertext Transfer Protocol
IE	Information Environment
IFSAR	Interferometric SAR
I/O	Input/Output
ITL	Information Technology Laboratory
JSU	Jackson State University
LSF	Load Sharing Facility
MPI	Message Passing Interface
MSF	Mass Storage Facility
MSI	Minority Serving Institution
MSRC	Major Shared Resource Center
NAVO	National Oceanographic Office
OKC	Online Knowledge Center
OOCORE	Out Of Core
PBR	Point-Based Representation
PDA	Personal Digital Assistant
PET	Programming Environment and Training
RCS	Radar Cross Section
RF-CTH	Reduced Functionality CTH
RGBA	Red Green Blue Alpha
RMS	Root-Mean-Square
RPC	Remote Procedure Call
SAR	Synthetic Aperture Radar
SHMEM	SHared MEMory
SOAP	Simple Object Access Protocol
SRTV	Soldiers Radio and Television
SSH	Secure Shell
UGC	Users Group Conference
VTK	Visualization Toolkit
XML	Extensible Markup Language

training schedule

For the latest on training and on-line registration, one can go to the Programming Environment and Training (PET) On-line Knowledge Center Web site:

https://okc.erdc.hpc.mil

Questions and comments may be directed to PET at (601) 634-3131, (601) 634-4024, or PET-Training@erdc.usace.army.mil

ERDC MSRC *Resource*
Editorial Staff

Chief Editor/Technology Transfer Specialist
Rose J. Dykes

Interdisciplinary Coordinator
Mary L. "Dean" Hampton

Visual Information Specialist
Betty Watson

ERDC MSRC HPC Service Center
Web site: www.erdc.hpc.mil
E-mail: msrchelp@erdc.hpc.mil
Telephone: 1-800-500-4722

The ERDC MSRC welcomes comments and suggestions regarding the *Resource* and invites article submissions. Please send submissions to the above e-mail address.

The contents of this publication are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products.

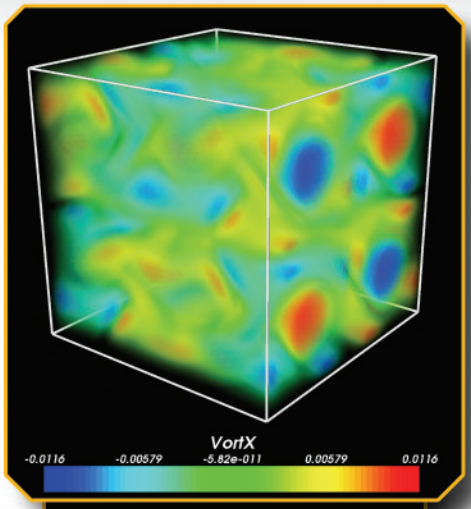
Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the DoD.

Design and layout provided by the Visual Production Center, Information Technology Laboratory, U.S. Army Engineer Research and Development Center.

Approved for public release; distribution is unlimited.

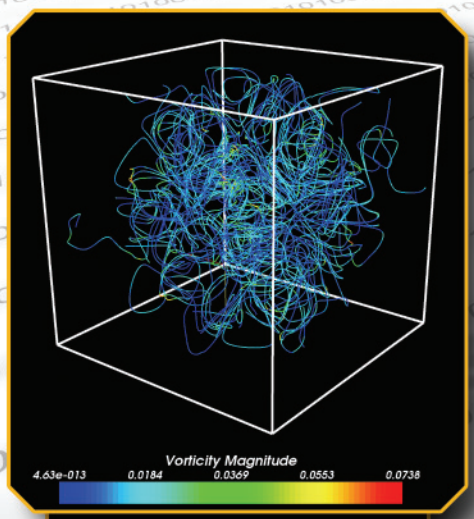
ezVIZ itIS!

Now in beta testing–available first quarter 2006



Volume Rendering

ezVIZ will take graphic visualization to a new level of *easy.* Offering a batch and soon-to-follow Web option, ezVIZ will provide the resources needed to rapidly deliver high-quality visualizations to the user's desktop.



Stream Lines

Isosurface of flow vorticity with underlying computational mesh.

The flow data were produced using a computational fluid dynamics algorithm developed by the Quantum Computing Group at the Air Force Research Laboratory, Hanscom AFB, MA

U.S. ARMY ENGINEER RESEARCH
AND DEVELOPMENT CENTER
INFORMATION TECHNOLOGY LABORATORY