Kitware released version 1.1 of the Kitware Image and Video Exploitation and Retrieval (KWIVER) open-source toolkit before the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Members of the computer vision team at Kitware appeared at the CVPR industry exposition to accent this technology and others. They also presented research, recruited for employment opportunities and served as conference chairs. Senior Director of Computer Vision Anthony Hoogs served as a general chair, and Director of Computer Vision Matt Turek served as a corporate relations chair.


Kitware posted links to these papers on the company blog, as well as entries on KWIVER. Kitware made the introductory release of KWIVER this year in January. KWIVER is a repository of open-source software for image and video analysis that includes tools for video stabilization, object detection and tracking, bundle adjustment, camera calibration, three-dimensional data reconstruction, super-resolution imaging and content-based image retrieval.

In September, we made a splash at the 20th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI). Senior Director of Strategic Initiatives Stephen Aylward co-led the organization of a workshop, Point-Of-Care-Ultrasound: Algorithms, Hardware, and Applications. We have several projects in the works that look to help medical professionals benefit from ultrasound systems. Numerous projects incorporate ultrasound probes from Interson, which identified us as an approved systems integrator for its probes. We have written many blog posts on ultrasound imaging, and we plan to write more, so please follow up at https://blog.kitware.com/tag/ultrasound.

**Editor’s Note**

This summer, we went to the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). It was wonderful to make many new acquaintances and to see some of the beautiful sights that surround the Hawaii Convention Center. Recruiting was one of our focuses at the conference and over the summer in general. Since our last issue, we brought in over 10 team members! To read their profiles, please refer to the Kitware News section of this publication. In addition, the news section has a story on Senior Director of Computer Vision Anthony Hoogs and his seat on the Information Science and Technology (ISAT) Study Group, which the Defense Advanced Research Projects Agency (DARPA) founded.

In September, we made a splash at the 20th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI). Senior Director of Strategic Initiatives Stephen Aylward co-led the organization of a workshop, Point-Of-Care-Ultrasound: Algorithms, Hardware, and Applications. We have several projects in the works that look to help medical professionals benefit from ultrasound systems. Numerous projects incorporate ultrasound probes from Interson, which identified us as an approved systems integrator for its probes. We have written many blog posts on ultrasound imaging, and we plan to write more, so please follow up at https://blog.kitware.com/tag/ultrasound.

**The TeleSculptor application in the Motion-imagery Aerial Photogrammetry Toolkit (MAP-Tk) extracts depth from aerial video. MAP-Tk is part of KWIVER.**
The release of version 1.1 enhanced KWIVER for use cases in conducting video surveillance and processing underwater images, among others; matured the build process; revised documentation; and better coordinated how various pieces of the toolkit work together. The Motion-imagergy Aerial Photogrammetry Toolkit (MAP-Tk) is one such piece. When Kitware first presented MAP-Tk at CVPR two years ago, it contained libraries of algorithms and structure-from-motion tools for video analysis. As MAP-Tk grew, its framework and core algorithms suited broader applications. To make them easier to reach, Kitware relocated these components inside KWIVER.

Kitware simultaneously turned the development of MAP-Tk toward specialized end-user tools. The primary tool, which the company now calls TeleSculptor, provides a graphical application for photogrammetry. In MAP-Tk 0.10, Kitware garnished TeleSculptor with support for carrying out a full structure-from-motion pipeline without receiving aid from command-line tools. Kitware released MAP-Tk 0.10 alongside KWIVER 1.1.

PARAVIEW 5.4 PREMIERES IN ADVANCE OF ISC HIGH PERFORMANCE

After previewing ParaView 5.4 in a series of blog posts, Kitware pushed the final version. Together, thirty developers added over 430 commits to the software.

“We revised the color legend with significant improvements in the choice and placement of graduations and annotations,” said Utkarsh Ayachit, a distinguished engineer and the lead developer of ParaView at Kitware.

Kitware team members called attention to ParaView 5.4 in a workshop at ISC High Performance 2017. The company timed the development cycle to acquaint conference registrants with release milestones:

• The Multi-block Inspector panel, through which users review and modify properties for blocks in hierarchical multi-block datasets, received a redesign with performance and usability in mind.
• The approach for loading state files in ParaView changed. The approach now allows ParaView to “Search files under specified directory.” This ability lets users share state files along with the datasets that these files need.
• Reformatted dialog boxes for saving screenshots and animations improved access to options that affect saved results, including color palette, background transparency and spacing between views.
• Axes Grid, a mechanism for annotating coordinate axes in three-dimensional views, extended to annotate each individual dataset.
• The CFD General Notation System (CGNS) file reader gained support for boundary condition patches for curvilinear grids and support for block selection according to family names.

• The number of supported VTK-m filters increased. Individually, these filters offer accelerated cell and point averaging, clipping, unused points removal and general surface extraction. In addition, the VTK-m contour filter acquired the ability to specify multiple isocontour values.
• Support improved for High Dots Per Inch (HiDPI) displays such as Retina.

“In the next release, ParaView 5.5, users can expect usability to continue to advance,” Ayachit said. “Specifically, the release will incorporate OSPRay path tracing for realistic lighting.”

VERSION 8.0 OF THE VISUALIZATION TOOLKIT ASSIGNS NEW CODE STANDARD

Kitware passed another turning point for the Visualization Toolkit (VTK) with the release of version 8.0. The release became the first to benefit from C++11 compliant compilers. VTK now officially supports different aspects of C++11 such as default constructors, static assertion, non-static data members and enumeration declaration.

“The new features in C++11 allow developers to be more productive and eliminate common sources of bugs,” said Dave DeMarle, a principal engineer at Kitware and a developer of VTK. “Now that VTK enforces the availability of a C++11 compiler, developers can rely on capabilities without maintaining awkward workarounds.”

For high-performance computing, the 8.0 release annexed the VTK-m framework of tools. These tools include new filters that process data. Kitware uploaded the filters to the Accelerators/Vtkm folder in the VTK repository.

Outside of VTK-m, the release merged algorithms that process points and geometries. One algorithm (vtkLagrangianParticleTracker) visualizes particles as they move through simulations, and another algorithm (vtkCookieCutter) precisely cuts a two-dimensional (2D) geometric surface with a separate 2D surface that acts as a stencil. Additional algorithms (vtkDensityPointCloudFilter and vtkUnsignedDistance) operate on point clouds. The release also augmented existing algorithms in VTK. The algorithm for dual depth peeling, for example, developed the ability to render volumes.

In addition, the release added the QVTKOpenGLWidget class, which provides a robust integration of VTK and Qt 5. The release also improved the OpenVR module, which pairs data with Oculus Rift and HTC Vive.

“The transitions to Qt 5, C++11 and Python 3 give users, developers and packagers a great deal of capability and flexibility in VTK-enabled applications,” DeMarle said.

VTK is an open-source software platform that manipulates and displays two-, three- and four-dimensional data. The VTK download page contains files for version 8.0. For more
points of the release, please read the Kitware blog. For assistance with VTK, please contact kitware(at)kitware(dot)com.

This technology was supported by the National Institute Of Biomedical Imaging And Bioengineering of the National Institutes of Health under Award Number R01EB014955. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

This material is based upon work supported by the U.S. Department of Energy, Office of Science, under Award Number DE-SC0012387.

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

This material was also supported by Sandia National Laboratories. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA0003525.

**VERSION 4.12 OF THE INSIGHT SEGMENTATION AND REGISTRATION TOOLKIT CENTERS ON PYTHON PACKAGES**

Kitware issued links to the 4.12 release of the Insight Segmentation and Registration Toolkit (ITK) on the ITK website. Python wheel packages served as a pillar of the release. With Python bindings, developers can use all ITK functions in Python.

“Accessible computational methods enable researchers to reproduce advanced algorithms and apply them to novel domains,” said Matthew McCormick, a principal engineer at Kitware and a developer of ITK. “Wheel packages can be quickly installed, and the Python programming language can be picked up without formal computer science training.”

Filters formed another cornerstone of the release. They furnished ITK with algorithms that improve the robustness and accuracy of image segmentation. The MorphologicalWatershed filter, for example, uses concepts from geophysics to section images. Two other filters calculate strain tensors on tissue. In May, McCormick wrote an article on these filters in the Insight Journal.

“Advanced, high-performance, N-dimensional algorithms are hallmarks of ITK,” McCormick said. “New filters for segmentation and registration complement the existing library well.”

Filter maintenance allowed ITK to display histograms of measurements more rapidly and to model shapes through principal component analysis with less memory consumption. In addition, the 4.12 release established more support for Microsoft Visual Studio, Clang and the GNU Compiler Collection. Code coverage also climbed to surpass the record that version 4.11 set. Kitware summarized code coverage and other aspects of the 4.12 release on its blog.

While the material discussed here is part of a community effort, at Kitware, this material is based upon work funded, in whole, by a $241,323 award from the National Library of Medicine.

**KITWARE POWERS PROJECT BUILDS WITH CMAKE 3.9**

Kitware officially completed the release cycle for CMake 3.9, which developers can now download. CMake consumed more than 900 commits from around 80 members of the development community since the last release happened in April 2017.

Numerous commits sought to relax constraints that affect parallel compilation with the Ninja generator. The relaxed constraints improve build times. Other commits targeted object libraries, which CMake can now install, import and export. These commits signaled the first steps of an iterative process that strives to make object libraries first-class citizens in CMake.

The release of version 3.8 previously made the CUDA programming language a first-class citizen in CMake. For the release of version 3.9, the CUDA_PTX_COMPILATION target property qualified CUDA to support PTX files in CMake and ship them as part of an application or a developer package. With this property, developers can perform just-in-time compilations of CUDA projects. Also for CUDA, the Visual Studio generator deepened support. Kitware elaborated on these and other particulars of CMake 3.9 on its blog.
Allegorithmic makes applications for texturing. Texturing is a virtual process that helps make a three-dimensional (3D) model appear realistic. With texturing, a 3D model of a wall can look like it is made of concrete. Sébastien Deguy, the CEO of Allegorithmic, describes texturing as “touching with the eyes.”

Four years ago, Allegorithmic launched the development of Substance Painter, which is shown in Figure 1. Substance Painter is an application that has helped Allegorithmic become a leader of the texturing industry. Major game studios now use the application. When Allegorithmic first created Substance Painter, the company only had a small team of software developers. They focused on a tool called Substance Designer. Substance Designer was originally developed in C++, using the Qt framework. It was built with QMake, which comes with Qt. The build process for Substance Designer was tailored over 10 years. Yet, it had a number of problems:

- Code was stored in a single, monolithic Subversion (SVN) repository that had a mix of libraries, in-house tools and application sources.
- Many handcrafted scripts were used to set up environment variables to specify items such as dependency locations, platform types and compilers.
- The build process required third-party libraries, which were manually maintained in a zipped file that each developer had to store on his or her computer. When the process incorporated a new dependency or compiler, the updated zipped file was manually deployed to all of the computers.
- Without a link between the source code and the various versions of third-party libraries, it was difficult to revert to earlier versions of the project build.
- The setup scripts were not cross-platform scripts. They required separate build and packaging processes for Windows and Mac OS X.
- Setting up new workstations—from code checkout to successful build—was a complex procedure that often took more than a day. This procedure heavily relied on detailed knowledge from developers.

Figure 1: Substance Painter adds texture to a 3D model of a fire hydrant.
DESIGNING A BETTER BUILD PROCESS
With the introduction of Substance Painter and the potential introduction of additional applications, Allegorithmic acknowledged that its cumbersome build process was a risk for the company. Looking toward the future, the company chose to form a new, more sustainable build process that could support new applications.

Developers at Allegorithmic wanted to easily check out source code, configure it, compile it and run it on their favorite platforms with their favorite integrated development environments (IDEs). They also wanted to maintain only one build file. Thus, the new build process needed to incorporate a fully reproducible and cross-platform build environment that could support Windows, Mac OS X and Linux without maintaining handcrafted, platform-specific scripts. The build process also needed to produce a stand-alone package that could be distributed without worrying about dependencies. The process had to be customizable, and it had to be able to generate documentation from source code, copy files at build time and use external pre-processing tools. In addition, the process had to easily handle third-party libraries. To accomplish this, the build system had to contain the correct versions of the libraries.

Since the software that Allegorithmic planned to develop would depend on Qt, QMake was initially implemented to achieve these objectives. Unfortunately, QMake did not suit the requirements due to its poor documentation and slow performance. Allegorithmic developers were unable to find a satisfactory way to use QMake to manage dependencies.

Allegorithmic next considered CMake as a candidate. Many of the third-party libraries that the company used were already compiled with CMake, and/or they provided CMake configuration files. Additionally, some of the developers at Allegorithmic had experience with CMake, and they were happy with the results. Allegorithmic found that CMake was able to fulfill all of its requirements. The company successfully implemented a new build process that used CMake.

MANAGING THIRD-PARTY LIBRARIES
Managing third-party libraries for a cross-platform project is a nightmare in C++. Each project uses different versions of compilers, different compilation options and different versions of libraries, which makes it complicated to maintain binary packages for each permutation. The solution is to build the libraries as part of a project to guarantee that the binaries are fully compatible with the project. These library sources are directly referenced in the source tree of the project, which ensures the availability of any given version of the project.

To link the library sources to a particular version of Substance Painter, Allegorithmic relies on the ExternalProject function of CMake. This function defines the necessary steps to download, configure, build, install and test external libraries. Allegorithmic only uses the steps to build and install the libraries. It employs Git submodules to download the appropriate versions of third-party sources.

It is not always possible to build third-party libraries as part of the project. Some libraries are proprietary, and they do not have source code available. Others cannot be integrated with the project source tree because of their respective sizes; these large libraries take too long to build. For such libraries, Allegorithmic employs a simple internal tool that allows pre-compiled binaries to be uploaded and downloaded from internal servers. This tool is used within CMake files.

It downloads the appropriate version of each library. The tool enables Allegorithmic to directly keep track of binary dependencies in the project source tree as it would any other source dependencies.

As a result of its flexibility, CMake can wrap calls to make their syntax similar to that of ExternalProject. The following call, for example, downloads the appropriate version of Qt for the target platform and adds its location to the path in CMake, where it can be found when needed.

InternalLibrary
alg_add_external_archive(ALG_QT
WIN32 qt-5.7.1-p2
APPLE qt-5.7.1-p1-c++11
CENTOS qt-5.7.1-centos-p1
UBUNTU qt-5.7.1-ubuntu-p1
}

ExternalProject can also download pre-compiled binaries through HTTPS or common source control options. The downside is that CMake installs ExternalProject at build time and not at the time that CMake is configured. This timing prevents the use of the find_library() feature in CMake. Accordingly, all third-party management occurs in a separate CMakeLists.txt file, which must be built and configured before the actual project is built and configured.

An alternative is to provide a superbuild whose last ExternalProject step is the real project. This way, ExternalProject downloads and builds the third-party libraries before the project itself, allowing the project to properly reference the libraries.

MANAGING DEPENDENCIES
The target system of CMake manages dependencies. Every executable or library that is built in a CMake project is a target. Each target defines the include paths, compilation options and libraries that CMake needs for a project build. Each target also defines the instructions for any other target dependencies. CMake can treat external libraries as targets, using the IMPORTED target feature. The ability to work only with targets makes CMakeLists.txt files clean and easy to follow.

The CMake approach to dependency management is less prone to error than the previous approach that Allegorithmic
used. Since the details of a particular external library are hidden inside of a target definition, developers do not need to worry if they forget a compile flag or an include path. The target system has proved important for Windows builds, and it has helped mitigate the limitations of the original build procedure.

On Windows, unlike on Mac OS X and Linux, it is not possible to specify a file location in the executable for dynamically linked libraries (DLLs). This means that a successfully compiled executable that relies on several DLLs cannot run unless those DLLs are either copied to the same directory as the executable or their locations are added to the PATH environment variable.

Allegorithmic solved this issue by writing its own introspection method, which uses properties on CMake targets. The introspection functions return the paths to the corresponding DLLs. A final post-build step was added to the executable target, which copies the DLLs to the same directory as the executable. While creating multiple copies of each DLL is not ideal, it serves as an acceptable compromise, as symlinks in Windows do not always work properly for certain levels of user permission.

Introspection is also used to generate stand-alone packages for projects with install() in CMake. The only issue with install() is that it cannot be used for IMPORTED targets. So, Allegorithmic had to create a workaround. Instead of directly calling install(), Allegorithmic created a custom command that can introspect a target as described above and generate the appropriate install() for each dependency. Thus, it takes only a single command to install a target and generate a stand-alone installation for that target. Since the install process in CMake is flexible, the custom command allows additional processes to run. Such processes fix RPATH on Mac OS X and Linux or generate symbol information for the crash reporting system.

CMake also provides a BundleUtilities module that can help generate a standalone installation of an executable. Allegorithmic has not tried to use this module yet.

LOOKING BACK AT THE MOVE TO CMAKE
The move from QMake to CMake initially faced resistance from some developers at Allegorithmic. In retrospect, it was the correct move. After Allegorithmic developed the framework for Substance Painter, it successfully ported Substance Designer to that framework, which allowed the development team for Substance Designer to grow. The framework also enabled the rapid startup of new products and teams that focus on software rather than on technical details of infrastructure.

Today, Allegorithmic has many software development teams, all of whom use the new framework. Developers are happy with the build system, as they no longer have to bother with the messy details of building and packaging C++ applications. The maintainable framework allows developers to continue to tackle challenges in the build environment, while they dedicate most of their time to improving their software products.

ACKNOWLEDGEMENTS
Thanks go to Allegorithmic (https://www.allegorithmic.com), which allowed this look at behind-the-scenes work. Thanks also go to colleagues, who worked on the efforts discussed in this article. Finally, thanks go to Stephane Guy and Eric Batut, who reviewed this article.

Alexandre Chassany is a senior software developer at Allegorithmic, where he has worked for seven years. He is highly versatile, and he is passionate about every aspect of software development—from software architecture to release processes. He is active on LinkedIn at https://www.linkedin.com/in/achassany and on Twitter through the handle @achassany.
Fully coupled multi-component and multi-scale modeling systems such as Regional Earth System Model (RegESM) are used to represent and analyze complex interactions among physical processes. A typical application of RegESM can produce tens of terabytes of raw data. This data determines the resolution of the spatial grid, the length of the simulation and the number of represented model components. The increased complexity of multi-component modeling systems results in extra overhead in disk input/output and network bandwidth. Thus, the systems require extensive resources for computation and storage.

Due to the increased complexity of multi-component modeling systems, the conventional post-processing approach has become insufficient to analyze and understand in detail fast-moving processes and interactions among model components [1]. In situ visualization has been used to overcome the limitations of the conventional approach. When compared to the conventional approach, in situ visualization can analyze key information that multi-component Earth system models generate in a higher temporal resolution. In addition, in situ visualization does not entail extensive code development and restructuring.

This article discusses in situ visualization and highlights work that tested its integration with RegESM. This work was presented in “Towards in situ visualization integrated model coupling framework for earth system science” at the Fourth Workshop on Coupling Technologies for Earth System Models [2].

**CO-PROCESSING AS PART OF REGESM**

In a conventional simulation system (Figure 1), ParaView Catalyst integrates a visualization pipeline with simulation code through an adaptor. This adaptor acts as an abstraction layer or a wrapper layer. Custom adaptor code is developed in the C++ programming language. It transfers information from the simulation code to ParaView Catalyst.

The new approach (Figure 2) aims to create a more generic and standardized co-processing environment for Earth system science. The approach integrates in situ visualization. In addition, the approach couples existing Earth system models with the Earth System Modeling Framework (ESMF) library [4] and the interface of the National Unified Operational Prediction Capability (NUOPC) Layer [5]. In the new approach, an adaptor interacts with an ESMF driver, which synchronizes the model components, the data exchange and the spatial interpolation that occurs among the computational grids of the model components.

Adaptor code defines the underlying numerical grid (structured or unstructured) and the associated multi-dimensional fields, using the Visualization Toolkit (VTK) [3]. ParaView Catalyst processes the data, performs co-processing and creates the desired final products. These products include rendered images, derived fields and added value statistics such as spatial and temporal averages.

The ESMF driver transfers the underlying numerical grid of each model component, or each source, to the co-processing component (Figure 3), or the destination. The transfer is facilitated by the ESMF library [4] and the interface of the NUOPC Layer [5]. The model components in Figure 3 are atmosphere (ATM) and ocean (OCN). While these individual components cannot interact directly with the adaptor, the adaptor code provides a seamless interface. It maps the ESMF field object (ESMF_Field) and the grid object (ESMF_Grid) to their VTK equivalents. To do so, it uses application programming interfaces (APIs), which are supplied by ParaView Catalyst and VTK.
TRANSFERRING COMPUTATIONAL GRIDS AND MAPPING TWO-DIMENSIONAL DECOMPOSITIONS

Due to the nature of Earth system modeling and its demand for extensive computational resources, Earth system models are designed to take advantage of parallel programming through Message Passing Interface (MPI). Based on the parallelization of model components, Earth system models use two-dimensional (2D) domain decomposition to solve a set of equations such as a set of Navier-Stokes equations. The computational grid of an individual model component and its 2D decomposition configuration are represented by `vtkMultiBlockDataSet` and `vtkStructuredGrid`, respectively.

A problem arises when loosely coupled visualization and modeling systems are considered. The model components and the co-processing component may run on different computational resources or in different MPI communicators (e.g., `MPI_COMM_WORLD`). As Figure 3 indicates, in ESMF convention, computational resources are assigned Persistent Execution Threads (PETs). As Figure 3 also demonstrates, the number of MPI processes in the model components and the number of MPI processes in the co-processing component may differ. If this is the case, the 2D decomposition configurations will need to be restructured. The co-processing component is responsible for modifying the 2D decomposition configurations of the numerical grids (Figure 4).

To allow the co-processing component to run in a specific MPI communicator, the `coprocessorinitializeWithPython` function includes support for the `int *fcomm` argument. The following Fortran code provides this argument.

```fortran
  g_coprocessor = vtkCPProcessor::New();
  MPI_Comm handle = MPI_Comm_f2c(*fcomm);
  vtkMPICommunicatorOpaqueComm*Comm = new vtkMPICommunicatorOpaqueComm(&handle);
  g_coprocessor->Initialize(*Comm);
```

After the co-processing component modifies the 2D decomposition configurations of the numerical grids, it passes them to the adaptor.

USING AN INTEGRATED SYSTEM TO ANALYZE HURRICANE KATRINA

To test the co-processing approach, ParaView Catalyst was integrated with version 1.1 of RegESM [6]. The state-of-the-art driver that is responsible for the orchestration of RegESM and the data exchange among the model components is mainly developed by Istanbul Technical University (ITU). RegESM can incorporate four different model components: atmosphere, ocean, wave and river routing. The test used two of these model components, atmosphere and ocean, to analyze Hurricane Katrina.

Hurricane Katrina was the costliest natural disaster and one of the five deadliest hurricanes in the history of the U.S. The storm is currently ranked as the third most intense landfalling tropical cyclone in the U.S. Hurricane Katrina was established on the coast of southern Florida as a Category 1 storm on August 25, 2005. It entered the central Gulf of Mexico and strengthened to a Category 5 storm on August 28, 2005.

To observe the evolution of Hurricane Katrina, a simulation was performed between August 27 and August 30, 2005. The atmosphere model component (from Regional Climate Model (RegCM)) was configured to have a horizontal resolution of 27 kilometers (170 latitude x 235 longitude) and 23 vertical sigma layers, which established almost one million grid points (Figure 5). The ocean model component (from Regional Ocean Modeling System (ROMS)) had a spatial resolution of three kilometers (655 latitude x 489 longitude) and 60 vertical layers, which established 19 million grid points.
points (Figure 5). To exchange data between the model components and the co-processing component, a coupling interval of six minutes was selected. The coupling time step was set to three hours.

![Figure 5: ParaView represents atmosphere and ocean model domains along with information about topography and depth.](image)

Figure 5: ParaView represents atmosphere and ocean model domains along with information about topography and depth.

Figure 6 shows an snapshot of the integrated analysis of Hurricane Katrina. Surface wind vectors (in meters per second) were provided by the atmosphere model component. They are indicated as solid arrows. They reveal large-scale circulation in the region. In addition to surface wind vectors, clouds were retrieved using a three-dimensional relative humidity field and a direct volume rendering technique. Measurements of the surface height of the ocean (in meters) and surface currents (in meters per second) were provided by the ocean model component. They show the response of the surface of the ocean to the hurricane.

![Figure 6: An integrated in situ visualization of Hurricane Katrina uses results from an atmosphere model component and an ocean model component. Results were taken every six minutes. Source: Turuncoglu [7].](image)

Figure 6: An integrated in situ visualization of Hurricane Katrina uses results from an atmosphere model component and an ocean model component. Results were taken every six minutes. Source: Turuncoglu [7].

The complex and non-linear nature of the hurricane requires advanced integration of the high volumes of data that come from the model components. Thus, the hurricane is challenging to analyze and study. Detailed representation of the hurricane necessitates the use of model components with very high spatial resolution. Although the low spatial resolution of the atmosphere model component gives insight into the formation and the evolution of Hurricane Katrina, it lacks detail of the vertical structure of the hurricane.

![Figure 7: Wind speed determines the colors of features and stream tracers. Image adapted from Turuncoglu [2].](image)

Figure 7: Wind speed determines the colors of features and stream tracers. Image adapted from Turuncoglu [2].

Due to the issue of numerical stability in the atmosphere model component, it is not possible to perform the simulation at a resolution greater than three kilometers. An additional non-hydrostatic atmosphere model component such as the Weather Research and Forecasting model, which is developed by the National Center for Atmospheric Research, can give a more detailed representation of the hurricane and its interaction with the ocean.

In addition to a general overview of the region, it is possible to analyze individual model components and features in greater detail. It is possible, for example, to extract the backward stream field from Hurricane Katrina (Figure 7).

![Figure 8: ParaView Catalyst displays measurements of the surface height of the ocean in meters and surface current vectors in meters per second. Image adapted from Turuncoglu [2].](image)

Figure 8: ParaView Catalyst displays measurements of the surface height of the ocean in meters and surface current vectors in meters per second. Image adapted from Turuncoglu [2].

It is also possible to analyze the evolution of the state of the surface of the ocean (Figure 8).

Furthermore, it is possible to investigate the vertical structure of the core of the hurricane (Figure 9).
CONTINUING DEVELOPMENT

The new approach enhances and standardizes the interoperability between simulation code and an in situ visualization system. The model-coupling framework that the approach employs analyzes the high volumes of data that come from multi-component Earth system models. The ability to analyze data in a higher temporal resolution will open new possibilities, enhancing knowledge of non-linear interactions and feedback mechanisms among model components.

The plan is to make the interface of the ESMF library more generic. This will allow adaptor code to be used by coupled modeling systems other than RegESM. The overhead of the in situ visualization component of RegESM is another important topic of future development. It will be investigated in a series of standalone and coupled model simulations and in various visualization pipelines. The results of the benchmark simulations will be used to improve the overall performance of RegESM.

In addition, future work will investigate a way to automatically assign the PETs that are used by the co-processing component to graphical processing unit (GPU) resources. This work will increase efficiency in hybrid computing systems that are configured with nodes, with and without acceleration support.

ACKNOWLEDGMENTS

This work was supported by a research grant (116Y136) from The Scientific and Technological Research Council of Turkey. The computing resources used in this work were provided by the National Center for High Performance Computing of Turkey under grant number 5003082013. The Quadro KS200 used to develop the prototype version of the system was donated by NVIDIA Corporation as part of its Hardware Donation Program.

Thanks go to Rocky Dunlap and Robert Oehmke from the National Oceanic and Atmospheric Administration (NOAA) Earth System Research Laboratory (ESRL) and Cooperative Institute for Research in Environmental Sciences (Cires), Gerhard Theurich from Science Applications International Corporation and Andrew Bauer from Kitware for their very useful suggestions, comments and support.

REFERENCES


Ufuk Utku Turuncoglu is an associate professor at Informatics Institute at ITU in Turkey. His main areas of interest are computational simulation of atmosphere and ocean model components, visualization with in situ techniques, process automation with scientific workflow systems and climate science. He is also interested in the design and development of coupled Earth system models.
Multi-modal virtual surgical trainers and planners that are equipped with interactive physics-based simulations are becoming common in curricula for medical training. They help to hone procedural and broader skills, thereby improving overall surgical outcomes. These simulators demand high frame rates and high-fidelity simulations [1]. To improve medical training, a team aimed to increase the fidelity of surgical simulations [1].

In medical training, adhesive contact may be observed between surgical tools and internal organs [1]. This contact is caused by adhesive forces [2]. Adhesive forces tend to oppose the relative motion of bodies under contact and result from material damage that occurs on a microscopic scale [3].

To observe the evolution of adhesive forces over time, the team modeled an elastic block as it fell under gravity from a preset height onto a rigid plane. Figure 1 plots the evolution of adhesive forces over time for different levels of adhesion stiffness. As the figure indicates, the forces fluctuated in response to the cycling of the internal energy of the block between potential (elastic) energy and kinetic energy. For lower adhesion stiffness, the elastic block did not detach from the plane upon initial contact. For higher adhesion stiffness, the adhesive forces did not prevent the block from detaching from the plane upon first impact.

The team built on a modified version of a previously known solver for the MLCP: iterative constraint anticipation (ICA) [6]. ICA can be employed in real time. The modified version on which the team built is described in Arikatla et al. [7]. For its work, the team proposed a new algorithm: iterative adhesive contact solver (IACS) [1]. Figure 3 outlines IACS.

![Figure 1: Forces of adhesion (normalized with body weight) evolve over time for an elastic block that falls on an adhesive plane.](image1)

![Figure 2: An elastic block undergoes various stages of detachment from an adhesive plane.](image2)

![Figure 3: Steps highlight IACS. Arikatla et al. [1] explains what the symbols represent. Source: Arikatla et al. [1].](image3)
IACS can adapt to various models of adhesion, and it can be applied to real-time simulations [1]. As an example, the team chose a model of adhesion proposed by Raous, Cangémi and Cocu [8]. In this model, the two states of bodies under contact, bonding (adhesion) and debonding (detachment), are characterized by a rate-dependent adhesion intensity (\( \beta \)) [8]. Gascón, Zurdo and Otaduy [9] simulated this model in a mechanical system. The approach in Gascón, Zurdo and Otaduy [9] treats adhesive forces in three orthonormal directions in the local contact frame of reference as unknowns at each contact point. At each time step, the inverse of the coefficient matrix of the system gets computed [9]. This results in a bottleneck with \( O(n^3) \) computational complexity. Thus, the approach has limitations when applied to real-time simulations [1].

Alternatively, to refrain from inverting the system matrix, IACS splits the unknowns into contact states (bonding or debonding), contact forces (adhesive and normal) and displacements [1]. This allows the adhesive forces to be treated explicitly. In addition, whereas the approach in Arikatla et al. [7] first estimates the states/forces in an approximate solver and then solves for the final state, IACS continually updates the states and forces until it undergoes convergence [1]. According to the convergence criteria, IACS stops when the change in the displacement field between consecutive iterations is less than a preset threshold.

It is important to note that the converged solution that is based on this criteria is not guaranteed to be the solution to the MLCP. The fact that the unknown contact states and adhesive forces are not implicitly formulated as part of the MLCP system makes it difficult to establish practical residual-based convergence criteria.

TESTING AND FUTURE WORK

As Figure 4 depicts, the team tested IACS on a case in which a rigid tool interacted with a liver [1]. The testing focused on adhesive forces; it did not consider frictional forces, which are generally coupled with adhesive forces.

Frictional forces can be incorporated in future work without discounting the present approach, as IACS can extend to models of friction. The team not only plans to incorporate frictional forces in future work, but it seeks to understand the theoretical guarantees that IACS can provide for convergence under certain assumptions of models of adhesion.

REFERENCES


NVIDIA INDEX™ PLUG-IN FOR PARAVIEW GETS UPDATED

NVIDIA Index™ is a commercial software that enables the use of graphics processing unit (GPU) clusters for real-time visualization of large volumetric and polygonal datasets. ParaView users became able to try Index™ for volume rendering in ParaView for the first time in 2015. Since then, new efforts have updated the plug-in to add support for volume rendering of unstructured grids.

A volume rendering of an unstructured grid uses NVIDIA Index™.

In addition, new licensing terms have made it easier to use the plug-in at no cost in several cases. More details on the latest Index™ update is available on the NVIDIA blog at https://blogs.nvidia.com/blog/2017/06/19/index-on-paraview.

KITWARE BRIEFS AT XPONENTIAL 2017

Kitware presented the brief “Linking Unmanned Systems, Visible and IR Video, Computer Vision, and Humans Together for Real-Time, Squad-Level, Battlefield Situational Awareness” at the Association for Unmanned Vehicles Systems International (AUVSI) XPONENTIAL 2017 conference. In the brief, Keith Fieldhouse, assistant director of computer vision, granted insight into support of squad-level activities from Kitware with contributions to unmanned systems, sensors and computer vision software. He discussed the intelligent integration of various platforms, sensors, software and humans to demonstrate the techniques, challenges and value that intelligent integration provides. The brief occurred Monday, May 8, 2017, from 4:30 to 5 p.m. CDT in room C140 of the Kay Bailey Hutchison Convention Center Dallas in Texas.
“Currently, our warfighters at the squad level do not have the tactical advantages available at the brigade level,” Fieldhouse said. “We are developing capabilities to give squads extra sets of eyes on the ground and in the sky to provide actionable intel in real time without overloading warfighters with additional data.”

KITWARE AND INTERSON PARTNER TO FACILITATE PORTABLE ULTRASOUND

Kitware became an approved systems integrator for Interson ultrasound probes. Kitware established years of experience with the probes through projects for medical imaging research, commercial consulting and open-source software development.

“Interson is very happy to feature Kitware as a trusted integration partner for our USB ultrasound arrays,” said Interson’s Director of Systems Integration Bill Wiedemann. “Like many others, Interson has relied on Kitware and Kitware solutions over the past seven years.”

For a Small Business Innovation Research (SBIR) grant from the National Institutes of Health (NIH), Kitware and its team used Interson probes to construct a proof of concept of a point-of-care ultrasound system. The team proposed the system to complement first responders as they locate internal bleeding on scene. The grant work applied a signal analysis algorithm that Kitware pioneered for tissue identification. Through follow-on work, the team intends to complete the system to intelligently assist first responders with probe placement, image interpretation and injury recognition.

“Point-of-care ultrasound systems that combine ultrasound probes with innovative machine learning algorithms and customized user interfaces will revolutionize in-field triage, in-home care and in-hospital monitoring,” said Stephen Aylward, Ph.D., the Kitware senior director of strategic initiatives. “Interson probes provide the portability and advanced imaging capabilities at the price point that these systems require.”

Aylward was a co-organizer of a workshop on point-of-care ultrasound for this year’s Medical Image Computing and Computer Assisted Intervention (MICCAI) conference. The workshop included presentations and demonstrations that covered a variety of topics such as in-field triage of traumatic brain injury and intuitive ultrasound guidance through augmented reality. To learn more about the workshop or Kitware point-of-care applications, please contact kitware(at)kitware(dot)com.

Portions of the research reported in this publication were supported by the National Institute Of Biomedical Imaging And Bioengineering and the National Institute of General Medical Sciences of the National Institutes of Health under Award Numbers R43EB016621 and R01EB021396. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

GPU TECHNOLOGY CONFERENCE OFFERS PARAVIEW LAB

Chintan Patel, the senior product marketing manager at NVIDIA’s Tesla Business Unit, discussed an opportunity to get to know ParaView in “ParaView Users – Bring Your Data to GTC, Gather Insights Like Never Before.” As he details in the post at https://blogs.nvidia.com/blog/2017/04/26/hpc-visualization-gtc, this year’s GPU Technology Conference (GTC 2017) featured a lab titled “Interactive HPC Volume Visualization in ParaView.” Attendees of the lab learned about NVIDIA IndeX™. While the lab had example data, attendees could bring their own to explore.

Robert Maynard, a principal engineer at Kitware, joined Patel at the lab. Maynard also took part in a talk titled “Build Systems: Combining CUDA and Modern CMake” at the conference, which took place from May 8 to May 11, 2017, at the San Jose McEnery Convention Center.

In addition to the “Interactive HPC Volume Visualization in ParaView” lab, ParaView was part of “HPC Visualization in Virtual Reality.” This conference activity was a “Connect with the Experts” session. It occurred May 9, 2017. During the session, NVIDIA overviewed a ParaView plug-in for virtual reality.

KITWARE COMPLEMENTS LEADERSHIP TRANSITIONS WITH PROMOTIONS

Kitware continued its transitions in team management and organizational structure with four promotions.

“This year, our offices in New York, North Carolina, New Mexico and France have undergone significant growth, particularly in data and analytics,” said Lisa Avila, the president and CEO of Kitware. “We are happy to recognize the leadership of several team members as well as the contributions of our entire company.”

Kitware recognized the leadership of Jeffrey Baumes, who the company promoted to director of data and analytics. Baumes joined Kitware in 2006, after he completed a doctorate in computer science. He has steered efforts such as
XD ATA and the Resonant software platform to fit industries that include defense, healthcare and energy. As director, Baumes will expand the software platforms and the technical strategy of the data and analytics team.

Stephen Aylward also started a new role as senior director of strategic initiatives. Aylward was senior director of medical research and senior director of operations in North Carolina. In 2006, he coordinated the startup of the Kitware office in this location. He has helped it to grow to over 40 team members and has guided several medical research efforts. In his new role, Aylward will plan and promote the trajectory of Kitware, fostering nascent technical developments and enriching synergies among Kitware software platforms and teams.

To further technical developments and synergies, Kitware named Andinet Enquobahrie director of medical computing. Enquobahrie has a doctorate in electrical and computer engineering as well as an MBA with a focus in technology evaluation and innovation. Since he joined Kitware in 2005, he has built and maintained relationships with collaborators, explored funding opportunities and led a team of research and development engineers to execute projects in image-guided intervention that influence fields from optometry to orthodontics. As director, Enquobahrie will guide the medical computing team as they continue to create algorithms and design software for academic researchers and commercial customers with the Insight Segmentation and Registration Toolkit (ITK) and 3D Slicer.

Kitware also made Matt Turek a director. Turek graduated with his doctorate in computer science and began at Kitware in 2007. He has worked with Anthony Hoogs, senior director of computer vision, to manage the computer vision team; increase its membership to more than 30; and maintain relationships with technical institutes, government agencies and leaders in satellite imagery. As a result of his ability to grow important customer bases, Kitware named Turek assistant director of computer vision in 2013. As director of computer vision, he will assume broader responsibility of the operation of the computer vision team.

D ARPA NAMES ANTHONY HOOGS TO ISAT STUDY GROUP

The Defense Advanced Research Projects Agency (DARPA) has named Anthony Hoogs to the Information Science and Technology (ISAT) Study Group for a three-year term beginning this summer. The group brings 30 of the brightest scientists and engineers together to identify new areas of development in computer and communication technologies and to recommend future research directions.

The ISAT Study Group was established by DARPA in 1987 to support its technology offices and provide continuing and independent assessment of the state of advanced information science and technology as it relates to the U.S. Department of Defense.

Earlier in this summer, Hoogs served as a general chair of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). CVPR is the premier annual conference for computer vision research. It had more than 5,000 attendees in 2017. At Kitware, Hoogs is the senior director of computer vision. He leads the computer vision team, which has more than 35 members, including 15 Ph.D.s.

NEW JOBS WEBSITE OPENS FOR BUSINESS


Along with the jobs website, Kitware greeted new team members. The company also embraced the return of three team members who formerly fulfilled internships. So far in 2017, over 15 interns have made a difference at Kitware.
David Owens
Owens joined Kitware as a systems administrator in Carrboro, North Carolina. He holds over 20 years of experience in information technology.

Bryan Garrant
The system administration team also added Garrant. He became a technical support specialist. While Garrant attended ITT Technical Institute, he focused on computer network systems.

Forrest Li
The medical computing team hired Li as an R&D engineer. He completed a graduate degree in computer science at the University of North Carolina at Chapel Hill.

Pierre Assemat
Assemat started a one-year internship in Carrboro. He studies electronics, computer science and robotics at École Supérieure de Chimie Physique Électronique de Lyon.

Jonathan Crall
Crall rejoined Kitware, where he previously interned. He currently pursues his doctoral degree in computer science at Rensselaer Polytechnic Institute (RPI).

Jason Parham
Kitware congratulated Parham, who rejoined the company. Like Crall, Parham previously interned with Kitware. He also pursues his doctoral degree at RPI. Parham’s graduate work regards wildlife censusing.

John Westbrook
Westbrook added his knowledge of recruitment and career guidance to the human resources team. He is a human resources generalist, and he is part of the Triangle Society of Human Resource Management.

Nandini Seshadri
The business development team brought in Seshadri as a proposal specialist. She has received recognition for her writing and her skills in debate.

Caroline LaFleche
LaFleche came to the computer vision team as an annotation specialist. Her background involves mathematics and computer science.

Adrien Beaudet
Kitware welcomed Beaudet as an operation support specialist. He brings knowledge of financial management to his position in compliance and contracts.

Thomas Hastings Greer
Greer moved from an intern to an R&D engineer in Carrboro. He wrote a blog post on his initial experience as an intern at https://blog.kitware.com/pythonic-callbacks-and-iteration-in-vtk.

In addition to providing readers with updates on Kitware product development and news pertinent to the open source community, "Kitware Source" delivers basic information on recent releases, upcoming changes, and technical articles related to Kitware open-source projects.

For an up-to-date list of Kitware projects and to learn about areas into which the company is expanding, please visit https://www.kitware.com.

A digital version of "Kitware Source" is available in a blog format at https://blog.kitware.com/the-source-newsletter.

Kitware would like to encourage members of its active developer community to contribute to "Kitware Source." Contributions may include a technical article that describes an enhancement made to a Kitware open-source project or successes/lessons learned via developing a product built on one or more Kitware open-source projects. "Kitware Source" is published by Kitware, Inc., Clifton Park, New York.