Introduction to ITK Online Course: Q&A

Q: Do you have any suggestions on using Python with ITK?

Q: How do you do segmentation with SimpleITK?
A: The segmentation filters are exposed as SimpleITK classes, and wrapped for the interpreted languages.

Q: ITK 4.2 removed OrientedImage, what is the migration path for existing code?
For the particular case of the OrientedImage, you can simply replace it with the itk::Image. We integrated the behavior of the OrientedImage directly into the itk::Image.

Q: It seems difficult to include a filter based on a condition. Do you have any recommendations on how to approach that?
A: I think it is possible to add or remove a filter in the image processing pipeline. Can you be more specific?

Q: To follow up, on a filter based on condition: the goal is to include a filter based on a boolean (in C++), but GetOutput() returns a specific image type that it is not possible to declare (picture an if statement with conditional filter inside of it)
A: I think it's better to avoid this situation by redesigning the pipeline or have two pipelines for two conditions. The input and output image types should be determined for each filter beforehand.

Q: Will SimpleITK become part of ITK releases?
A: SimpleITK has its own releases, but they are typically synced with ITK releases.

Q: ITK requires lots of include directories to be specified (one for every module). Is there a way to reduce that (like boost)?
A: If you install ITK, the installed include directory is a single directory (ITK-4.1. e.g.) that has all the headers in a flat way.

Q: Follow up on includes: I’m not sure if the answer is correct, as even though the directory is flat, ITK own #include directives do not mention directories, so all module path have to be added as '-l' option.
A: It depends on whether you use ITK from the Build directory directly or whether you Install ITK. If you use ITK from the build directory, then you are right, you still have to add all the -l modules directories that you are using. But, if you Install ITK, and use it from the installed directory, then all the ITK headers are in a flat directory structure and you only need one -l entry (plus some others for VXL directories),

Q: What is the status/roadmap of filters running on GPU (there is a wiki, but there doesn’t seem to be support in the 4.2 release)?
A: They should be released with 4.2 by the end of June.
Q: What is the recommended way to find good initializations for registration? Is there any GUI that can be used to take care of flipping, etc.?
A: It depends on what kind of registration methods you are using. You could flip images using ITK classes. For viewing image or reorient images, I recommend using ITK-Snap.

A: ITK does not provide GUI applications. You may find it useful to look at the following applications: Slicer, VV, Volview, V3D, among others. That said, if you are using ITK registration, the first thing you want to do is to run the optimizer for zero iterations to see where the image is starting. You also want to include a command observer to monitor the progress of the registration.

Q: Will sampling change the output resolution of the image?
A: When doing resampling you must specify the resolution (pixel spacing and number of pixels) of the output image. So, yes, resampling in general will change the resolution of the output image. Of course, if you want to maintain the resolution of the output image to match the input image one, you can simply specify this in your code.

Q: In terms of performance, is ITK competitive with other commercial software for Image registration? Do you have performance charts?
A: We don't have specific performance charts, but it's a good idea to give a try. There are some MICCAI challenges regarding different image registration methods in terms of performance both in accuracy and speed. You might find them interesting.

Q: How can I customize Deformable Image registration 2 example in ITK to use different metrics other than the histogram matcher?
A: You should be able to instantiate a new metric and simply set the registration method to have the new metric.

Q: I have a university project on deformable image registration using DEMONS (deformable Image registration 2 example in ITK), and would like to change the metric from histogram matcher to Mean Square difference. How can I do that?
A: Try instantiating MeanSquareDiference and assigning it to the registration method.

Q: Where can I get further assistance, if I have more questions?
A: The mailing list is a great place to start for questions. There is a lot of expertise, and the community is one of the best resources for ITK.

Q: Is there any easy guide on how to modify ITK image registration classes to utilize GPU?
A: Not at this point. A set of GPU support classes will be included with ITK 4.2, but not for image registration. On the other hand we are collaborating with the UCL team that developed NiftyReg, which is a GPU-based image registration library, and we will facilitate that you can use NiftyReg as an ITK module. The GPU work is still in flux/under development, though there should be documentation and wikis with examples coming out as the GPU modules stabilize.

Q: Where can I find the recording of this webinar?
A: The recording of the webinar will be available here: http://itk.org/ITK/resources/webinars.html. It will also be available on http://www.kitware.com/products/webinars.html