Appendix for:

Nondestructive high-resolution visualization and measurement of anisotropic effective porosity in complex lithologies using high-resolution X-ray computed tomography

Richard A. Ketcham[1,*] and Gerardo J. Iturrino[2]

[1]Jackson School of Geosciences, The University of Texas at Austin, Austin, TX, 78712, [2]Lamont-Doherty Earth Observatory of Columbia University, Route 9W, Palisades, New York, 10964-8000, USA.

Submitted to Journal of Hydrology, October 2003

Appendix: IDL routines for alignment and CDF generation

The alignment procedures used in this paper are implemented in the IDL programming language (Research Systems, Inc., Boulder, Colorado; http://www.rsinc.com/idl/). IDL runs on many platforms (Windows, Macintosh, Linux, UNIX), but requires an IDL license. These routines have only been tested on the Windows version of IDL, but should run on any platform.

The principal computer configuration requirement for running these routines is that it has enough memory (RAM) to hold both data sets (dry and wet) in memory at the same time. For example, if each data set consists of a stack of 512 files, each of which is a 512x512 16-bit TIFF image, one data set requires 256 megabytes of RAM, and thus two require 512 megabytes, in addition to that required for overhead from the operating system and IDL environment. An additional complication is that the Windows operating system is very prone to fragmenting memory, which prevents IDL from allocating the contiguous blocks of space it needs; thus a data set may not load even if there appears to be sufficient memory. The only solution to this problem at this time is to have memory far in excess of that which appears required. Also, it may help to close all other applications and reboot the computer to "clean up" memory immediately before attempting to run these routines.

Using these routines successfully requires a basic working knowledge of IDL and how procedure calls work. A very brief overview is provided here, but more complete information can be found in the IDL documentation. IDL commands are entered at the command prompt that is available when IDL is running. The basic syntax for running an IDL command is:

IDL> PROCEDURE_NAME, *Argument* [, *Optional Argument*]

Where "IDL>" indicates the command prompt, PROCEDURE_NAME is the name of the procedure, *Argument* is a required parameter, and *Optional Argument* is an optional parameter. All arguments are specified by commas. There are two types of arguments: positional parameters and keyword parameters. Positional parameters are used by passing a constant or variable, and are specified by their relative position after the procedure name: the first positional parameter is the first one listed after the name, etc. Positional parameters can be required or optional, but using any positional parameter requires using all before it; for example, using the third positional parameter requires using the first two as well. Keyword parameters have the syntax

KEY_PARAM_NAME=*param*, where KEY_PARAM_NAME is the name of the keyword parameter, and *param* is a constant or variable. Keyword parameters are usually optional, and can be entered in any order.

To use these routines, start IDL and open and compile the source code file ALIGN3D.PRO. The two procedures in this file intended to the called by the user are Align3D and A3D_PorosityCDF, which respectively perform alignment and CDF generation. Descriptions of each routine are provided below.

Align3D

Purpose

Aligns two nearly-aligned data sets, and interpolates one so it matches the other as closely as possible. This implementation is based on a sample being scanned twice, once "dry" and the other infiltrated with a fluid ("wet"). It attempts to find a series of tranlational and rotational displacements: (dX, dY, dZ, phiX, phiY, phiZ) that maps the "wet" data set to the "dry" one. Translational displacements are in voxels and rotational displacements are in degrees. If the SCALE keyword is set, a seventh parameter is added (dS) that allows the "wet" data to be uniformly downscaled to compensate for expansion.

The routine assumes a certain directory and naming structure to streamline the process. The starting point is a root directory with the sample number as its name, and subdirectories containing the "dry" and "wet" data having the same name with "d" and "w" appended, respectively. The data files should be in TIFF format with ".TIF" extensions, and be in alphabetical order.

An example directory/filename configuration:

c:/files/17r
  c:/files/17r/17rd
    17rd0001.tif
    17rd0002.tif
    ... (etc.)
  c:/files/17r/17rw
    17rw0001.tif
    17rw0002.tif
    ... (etc.)

Positional Parameters

numdel: Number of border pixels to omit from consideration. Default=10.

Keyword Parameters

CORREL: Set to 1 to use correlation as image-fitting merit function. Otherwise minimizes the number of large outliers.

POINTS: Set to use point fitting, based on 2^{points} points.

IMAGE_INDS: For image fitting, set to a 2-element array to specify slice numbers to use. Default is 3*numdel from each end of stack.

INITGUESS: FltArr(6) with initial guess: [dX, dY, dY, phiX, phiY, phiZ]; if SCALE is set, it's a FltArr(7) with the seventh paramater being dS.

ADJUST_FINAL: Do a rotational adjustment on each interpolated image.

WET: Pointer to "wet" image stack; set to skip reading, or to return stack.
DRY: Pointer to "dry" image stack; set to skip reading, or to return stack.
FINAL_COEFS: FltArr(6) (or 7 is SCALE set) of final coefficients. Set to just regenerate the interpolated images.
SHOW: If image fitting is used, setting this will show each attempted fit.
COMBO_MEDIAN: Set to bin width of median filter to smooth right side of "combo" images. Default=3; =1 turns filter off (=0 does not!)
SCALE: If set, allows scale to be adjusted to account for uniform expansion of the wet material. Set to a value between 0.9 and 1.0 as an initial guess for the fractional contraction to scale the wet images back down to the dry.

Outputs

Creates three directories with the interpolated wet images, the difference images, and compressed "combo" images showing scan and difference data side by side. Also writes a file "ALIGN3D.LOG" with fitted parameters. If WET and/or DRY keyword parameters specified and they point to empty variables, these variables are assigned to pointers holding the appropriate data set. Freeing these pointers (using Ptr_Free) is then the responsibility of the user.

Example

```
; Align using a 10-voxel buffer, store dry and wet data sets in variables d8 and w8,
; respectively, if they're not there already; use the correlation image fitting method;
; include a final adjustment for rotation errors; and show fitting as it is occurring.
Align3D, 10, DRY=d8, WET=w8, /CORREL, /ADJUST_FINAL, /SHOW
```

Notes

This routine only works if the two data sets are fairly close to each other, or if the initial guess places them close to each other. "Fairly close" is roughly within 3 slices of each other and within 10 pixels and and 10 degrees of each other in any slice, and less than 2 degrees of tilting (if a wider range is desired the "range" variable can be reset, although a larger range makes it more likely that a convergence point will not be found).

If a good initial guess is required, place a break point at the line marked below with "***", and run the routine using image fitting, and the SHOW keyword set. Then, when the program reaches the break point, run a series of commands: print, A3D_TransFunction_Img(dX, dY, dZ, phiX, phiY, phiZ)changing the parameters as necessary until the data sets seem fairly close to each other. Then, set the variable "initGuess" to the vector of coefficients you've found and continue execution. If you want to use an alternative fitting method, after you've got your initial guess using the above method you can terminate execution of the routine and run it again using the INIT_GUESS keyword to set your starting point.

The alignment procedure is not able to cope well with a case of strong misalignment between data sets without having a reasonable initial estimate as to what the fitted parameters will be. In the routines provided with this paper, the defaults range of values inspected is ±10° for θ_z and ±2° for θ_x and θ_y, and ±3 voxels in *dz* and ±10 voxels in *dx* and *dy*. These ranges can be expanded by editing the source code, but having overly large ranges can prevent convergence to a good answer.

A3D_PorosityCDF

Purpose

Creates a CDF of % volume vs. % porosity given an image stack showing partial porosity.

Positional Parameters

fluidGS: Grayscale value for 100% porosity; use 16-bit value minus 2^{15}

outfile: Path+name for CDF file

Keyword Parameters

CDF: Set to named variable to return CDF.

PHI: Set to named variable to return phi (porosity).

DIFF: Difference stack; set to use or receive

MEDIAN: Pre-smooth the image data, using box median filter of width <med>

FIT_POROSITY: Hard-wires the net porosity represented in the CDF.

CORRECT_SHIFT: If FIT_POROSITY is specified, setting this parameter causes the source of error to be interpreted as a net grayscale shift between the two data sets, probably caused by beam drift, and the new drift is estimated. Otherwise, the variable fluidGS is adjusted.

Outputs

Keyword variables CDF and PHI are filled, if specified. Creates a text file with results.

Example

```
; Calculate CDF for sample 7f, assume df(H2O) is 4279
A3D_PorosityCDF, 4279, 'c:\files\7rcdf.txt'
```