

Introduction of Scale Factor to Multi-Resolution Registration

Jan Owoc

Final Report
Markin-Flanagan USRP
Fall 2005/Winter 2006

Abstract

Introduction: Registration is the process of finding the transformation that best maps the points of one image, called the moving image, to another image, called the fixed image. There are three main kinds of registration: point-based, surface-based, and volume-based. Volume-based registration has the potential to be the most accurate. Registration can be sped up by performing registration at a lower resolution, followed by registration at a higher resolution. This process of increasing the resolution during registration is known as multi-resolution registration. Registration is called rigid when the transformation sought involves only rotation and translation. In many cases it is useful to have the option of a scale factor.

Methods: The Insight Segmentation and Registration Toolkit (ITK) is a free set of program components that can be pieced together to create a registration program. Kuhn and Moser developed a rigid registration program that has been used successfully for many years. Due to limitations within ITK, it was impossible to simply extend the registration to include a scale factor. An entirely new program was written to perform multi-resolution registration with a scale factor.

Results: The new program is able to perform registration of micro-CT data stored in Scanco AIM files. The process involves picking points (landmarks) to be used as an initial guess, decomposing the transformation into transform parameters, writing a configuration file, and running the final registration. The program has been verified to detect scale differences correctly.

Discussion: Registration involves several interdependent components that are selected independently and must work together to be successful. Two of these components are a metric, which determines how close the fit is between the images, and an optimizer, which determines in what direction and by how much should the next guess be changed. The metric was chosen to be Mattes Mutual Information, and the optimizer was chosen to be Gradient Descent based on previous research. Testing of other metrics and optimizers is warranted given more time.

Conclusions: The new registration program performs multi-resolution registration correctly, including the introduction of a scale factor. It is possible to prevent the scale factor from changing, meaning this program serves as a replacement for the previous rigid registration program.

Introduction

Registration is defined as the process of finding the spatial transformation that maps points from one image to the corresponding points in another image (pg 239 of [1]). In medicine, there are three main applications of registration. The first is if a single patient is scanned multiple times and one wishes to track the changes in bone or tissue over time. The second is during the comparison of the scan of a patient to a scan of another patient. The third application is when a single patient is scanned in several different machines (eg. CT and MRI) and one wishes to use both data sets simultaneously.

In all cases, a patient is placed in a data acquisition machine and scanned. Regardless of the care taken, it is impossible to place a patient in the exact same orientation for a second scan. In addition, if the machine is not the same one, other parameters, such as offset or scale, may be different. The goal of registration is to find what transformation can be applied to one of the images, called the moving image, to make it line up with another image, called the fixed image. Registration is a critical step if one wants to be able to perform any subsequent analysis when comparing images.

Approaches to the Data

Given an image, be it two dimensional or three dimensional, there are three possible approaches to register it to another image. The first is known as the “landmark approach” or as “point-based methods.” This method involves the identifying of features, called landmarks or fiducial points, that are known to be always present in the image. These landmarks may be markers placed on the subject and appearing in the image or they may be recognizable features of the image itself. Based on the location of these landmarks, the appropriate transformation can be found quickly and easily. Such an algorithm is only as accurate as the detection of the markers or landmarks. In the case of markers, one has to physically attach something to the object being scanned. This involves human interaction and therefore introduces variability and irreproducibility. In addition, the identification of landmarks is frequently done by a human. The transformation is only as accurate as the person who is identifying and marking the points. Doing this properly requires considerable expertise, experience and time. Landmark registration is rarely consistent or reproducible and is not acceptable in many cases.

The second possible approach is known as the “segmentation approach” or as “surface-based methods.” This method involves segmentation of the original image. Segmentation is the process of converting an image into predetermined regions. There are many ways to segment an image. For example, in a CT image, an image could be segmented into “bone” and “not bone” based on a predetermined threshold. Because the registration algorithm will deal with the segmented image, the segmentation needs to be accurate. The main advantage of working with segmented data (and operating on the boundaries) is that it is often faster than using the original grey-scale values. The assumption is that if the (properly segmented) surfaces match, then the volumes themselves have also been successfully registered.

The third approach is to use the original data and try to correlate it directly. This is known as the “voxel intensity approach” or as “intensity-based methods.” As this approach uses the original data and not just the segmented surfaces or captured points, it is theoretically possible to obtain the best registration. Most of the time, the original dataset is significantly larger than segmented or landmark-picked datasets, and this method is ill-fated to be slower and to require more memory. However, due to the potential accuracy of working with the original data, the voxel intensity approach is often used as a “gold standard” for assessing the accuracy of other methods.

Transform Types

There are several common transforms used in image registration. The choice of transforms

depends on the specific application. The possible transform parameters are translation, rotation, scales (isotropic or anisotropic), shear, and deformations. A transform is a specific combination of one or more transform parameters. The simplest transform is the “rigid transform;” it is used when it is known that the objects to be registered are exactly the same size and shape, and the moving image is “rigidly” transformed to match the fixed image. Rigid registration involves only translation and rotation to align the images. If additionally it is known that the objects to be registered differ in size, a “similarity transform” can be used. The similarity transform uses isotropic (same in all directions) or anisotropic (different in each direction) scaling in addition to translation and rotation. A case where anisotropic scaling may be needed is when registering MRI to CT, where MRI does not necessarily have the same spatial resolution in all three dimensions. The most complicated of the traditional transforms is the “affine transform” which involves adding shear to a similarity transform. Deformable registration (where the image is allowed to “deform” in certain ways under specific circumstances) is beyond the scope of traditional registration, but is the topic of most of the current papers in the field of registration. Deformable registration would be used, for example, when mapping soft tissue which changes its shape.

Components of Registration

Registration is usually an iterative process with several components. In the case of the point-based approach, there is normally no iteration; the transform is calculated directly from the spatial coordinates of the landmarks or markers. In the case of the surface-based or volume-based registration methods, the components are the interpolator, metric, optimizer, and transform:

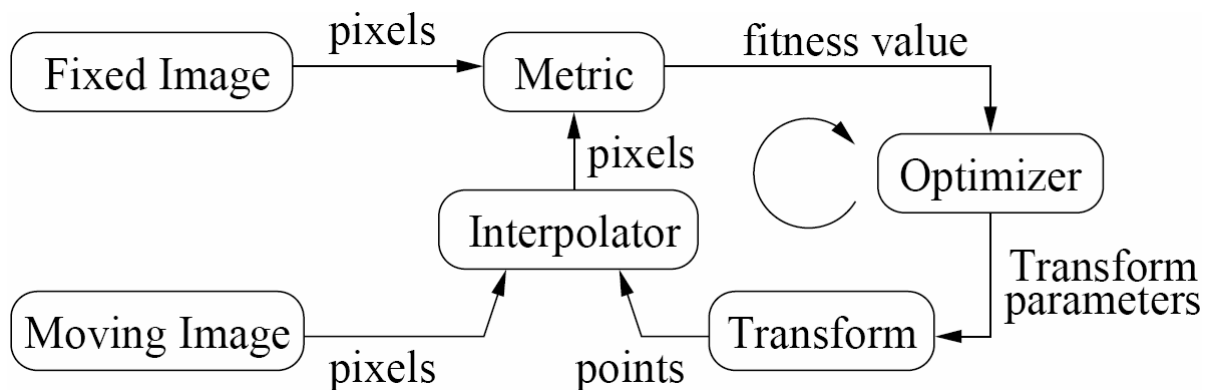


Illustration 1: from pg. 243 of [2]

The interpolator is so called because its job is to interpolate the image to approximate values of the image data at points not present in the original image. After a transformation, seldom do points from the moving image coincide exactly with the points from the fixed image. The simplest interpolator finds the point that is nearest to the desired location and uses the value of the voxel at that point. This approach is named “nearest neighbour” and is the fastest and the easiest to implement and use. A slightly more advanced method takes the weighted mean of several nearest neighbours in an attempt to interpolate between the neighbouring voxels, and is called “linear interpolation.” The most advanced method is fitting a multi-dimensional spline function to the nearest points, with the B-spline being the most common. This operation needs to be performed on every voxel after each iteration, and will be a bottleneck for speed. Although using the B-spline interpolator has the potential to yield the most accurate registration, it is significantly slower and its use may not be justified.

The “metric” returns a single “fitness value.” Using the interpolated points, the metric determines how closely the moving image coincides with the fixed image. There are about half a dozen metrics in common use, each with its specific purpose. One of the simpler metrics is the method of “least squares.” For images that have been registered, the sum of squares of the differences in image

intensity should be as small as possible. This method assumes that the images have the same intensity range. Another possible metric is to perform “normalized autocorrelation,” where the sum of the products of the respective voxel intensities is to be maximized. This method offers a slight improvement over least squares in that it also works when the image intensities are not the same. Other methods include “difference intensity” and “mutual information” and are explained in [1,2,3,4]. In the case of a segmented image, the metric is often of a different nature. Instead of iterating over the volume, the metric iterates over the moving image and tries to minimize the distance to the fixed image. The most complicated part of such an algorithm is finding which point on the fixed image is indeed the closest one. There are three algorithms in use: “head and hat,” “distance transforms” and the “iterative closest point algorithm.” The last of the three is a brute-force exhaustive search for the smallest distance. Although often faster, the algorithms dealing with segmented images are quite complicated.

The “optimizer” determines how to adjust the transform so the metric returns a better value. In the case of the four metrics described above, it is possible to compute the derivative of the metric with respect to the transform. Using something similar to a secant method, the optimal transformation can be found. The fourth component is the transform itself. The optimizer decides how to adjust the transform for the next iteration. The optimizer is closely linked to the transform and sometimes to the metric. This is because a change of one unit in terms of scale is much more significant than a change of one unit in terms of translation. If the adjustment is small, it is considered that the solution converged.

Using Multi-Resolution Registration

A critical part of registration is to provide an initial guess at the transformation. In order for the solution to converge, the transformation must be within a certain “capture range” of the desired result. This capture range is generally on the order of 10 degrees error in rotation and about 10% error in scale or translation. The result of registration of artificial data usually finds the correct transformation to within a single voxel.

During the first few iterations, the optimizer looks for a transformation that is relatively close to the correct transformation. The step sizes are rather large, and the registration is rather inaccurate at this stage. It is wasteful of computer resources to perform these first few iterations of registration on the entire image. It is possible to reduce the resolution several times for the first few iterations, and increase the resolution once the solution has started to converge. This idea forms the basis for multi-resolution registration.

The images are sampled down to lower resolutions, often several times, into “image pyramids.” The images can be first registered at $1/64^{\text{th}}$ resolution, followed by $1/8^{\text{th}}$ resolution. If the image is large, an additional $1/512^{\text{th}}$ resolution image could be created and added to the pyramid. Quite frequently, the image is properly registered at the lower resolution levels, and few iterations at the $1/8^{\text{th}}$ resolution are necessary, and even fewer at the full resolution.

A graphical representation of “image pyramids” is shown below.

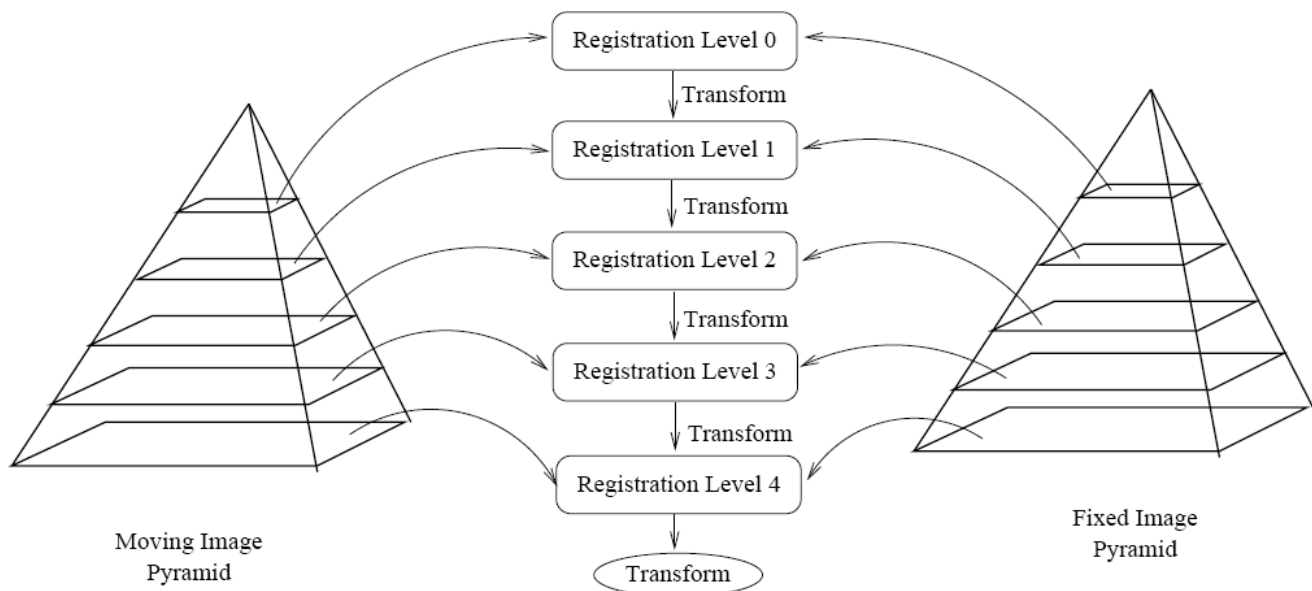


Illustration 2: Image taken from [3]

The purpose of this project is to allow the transform during multi-resolution registration to include an isotropic scale factor.

Methods

Existing Software

The National Library of Medicine has funded the creation of a set of program modules that help do routine tasks with images, including image registration. These program modules are collectively called the Insight Segmentation and Registration Toolkit (ITK), and have been used for many years at the University of Calgary for analysis of micro-CT data. ITK is an open-source, free, toolkit which has most of the code necessary to perform registration implemented in C++ using templating to be as portable and universal as possible. As such, it needs to be adapted to a specific purpose.

Programs that use ITK can be written in any of several languages, including C++. The programs implement a pipeline where the input of one filter connects to the output of the preceding filter. At the end of the pipeline, an “update” command is issued, and the pipeline executes the necessary commands to produce a useful output. Registration is done by setting up a registration class, connecting it to the appropriate inputs, interpolators, transforms, metrics, and optimizers, and then connecting the output to a file writer.

Three years ago, a rigid multi-resolution registration program was implemented by Michael Kuhn and Stephan Moser. Their work is documented in [5]. Kuhn and Moser showed that ITK is mature enough to implement a registration framework. This was not just a proof of concept, but also served as an excellent starting point for the introduction of a scale factor.

Attempt at Expanding Existing Software

Because there was a complete and working framework for rigid registration, it was a seemingly trivial task to extend the program to include isotropic scaling. The existing software used a quaternion rigid transform during registration. A quaternion is one of several ways of expressing rotation in 3-d. The quaternion has four components – three angles which define the orientation of the rotation vector,

and a fourth angle which defines how much the object is to be rotated. A detailed explanation of quaternions can be found in [6]. This transform was chosen because it had an optimizer that worked well with this transform, whereas the other, more generic, optimizers did not.

ITK does not have a class which lets the user use a quaternion rigid transform in combination with scaling. If one wishes to implement isotropic scaling, an Euler 3-d transform needs to be used in combination with scaling. An Euler 3-d transform uses three Euler angles, angles by which the object is rotated about the x, y, and z axis. Although there is a 1:1 correlation between Euler angles and a quaternion, constantly converting between the two is tedious, and in many instances impractical. In addition, the optimizer was one that worked with the quaternion rigid transform only, and it was impossible to adapt the optimizer to work with an Euler 3-d and a scale factor. Within a few weeks it became apparent that modifying the existing framework to include a scale factor was not going to work regardless of the effort put in, and registration needs to be reimplemented from the start.

The Development Process

The existing framework for rigid registration was abandoned in favour of a simpler model that could be easily adapted to future needs. When Kuhn and Moser were writing the registration program, ITK was not a mature technology and had many problems that needed to be worked around. Most of the code was written to overcome deficiencies or bugs in ITK. Over the last two years, ITK has become more stable and many features are implemented correctly that were not present at the time of the original program. Whereas the registration program was previously split into a dozen sub-programs to take care of various aspects of the registration manually, most of the control over the registration process could now be left in the hands of ITK itself. ITK has a steep learning curve, and even veteran programmers need a lot of examples to get going. Knowing this, the authors have provided nearly a hundred examples on how to do various things, including about a dozen examples of possible registration programs. The example known as `MultiResolutionRegistration2` was used as a starting point, as its structure was most similar to the desired final product.

The first step in registration is to read in the data to be registered. The micro-CT scanners used at the University of Calgary are manufactured by Scanco Medical, and the data comes out as a Scanco AIM. A Scanco AIM reader and writer were implemented many years ago to work with the Visualization Toolkit (VTK) framework. The VTK framework and the ITK framework are quite similar, but they do not share an internal image format. Kuhn and Moser had written an interface to read AIM files into ITK-compatible images, and it was found to work properly. The images were read in from the files and converted to the ITK format.

The second step in registration is to decide what transform, interpolator, metric and optimizer to use. Kuhn and Moser had found that creating a configuration file that is read in by the program is a convenient way for the end user to choose his or her preferred parameters. Due to limitations within C++, the end-user needed to have his or her choices restricted to avoid the program getting too large to handle. The only transform in ITK which matched the design criteria was the “Similarity 3D Transform,” and there was no need to implement the dozens of other possible transforms. Kuhn and Moser [5] had experimented with various interpolators and found that the linear interpolator gives the best trade off between accuracy and processor time; there was no need to implement nearest neighbour or B-spline. There are three metrics that have been used with micro-CT data: mean squares, normalized correlation, and Mattes Mutual Information. Through experimentation, Kuhn and Moser found that Mattes Mutual Information, although slightly less accurate, was significantly faster than the other two and therefore deemed the more appropriate choice. The user is able to additionally select parameters affecting the speed and accuracy of the registration. The two parameters in Mattes are the number of

histogram bins (more, up to about 50 leads to a greater chance of the solution converging) and the number of spatial samples (more, up to about 20% of the image size results in more accurate registration). The choice of optimizer turned out to be the most difficult. Kuhn and Moser had chosen to use the quaternion rigid gradient descent optimizer because it was tailored specifically to the quaternion rigid transform, and was one of the few optimizers correctly implemented at the time. Because the transform was changed to a similarity transform, this optimizer could not be used. Of the half-dozen possible optimizers, there were two good candidates: regular step gradient descent optimizer, and gradient descent optimizer. The regular step optimizer operates much like the bisection method of finding the zero of a function; it rotates, translates, and scales the image using a decreasing step size until it finds the optimal transformation. The gradient descent optimizer behaves like the Newton-Raphson method of finding the zero of a function; it adjusts the transform parameters according to the gradient and the user-specified learning rate. Kuhn and Moser [5] developed a clever algorithm for determining the learning rate based on the direction of the gradient, where the learning rate is decreased by 25% if the direction of the gradient switched, or increased by 10% if it did not. Through experimentation with many datasets and starting parameters, it was determined that the gradient descent optimizer was the better of the two for most micro-CT data.

The third step is to run the registration and to output the data. This was done by calling an update on the pipeline, and letting ITK take care of the rest. The moving image was then transformed to the orientation of the fixed image. This image was converted to a VTK image and written out to a Scanco AIM file for further use.

Results

Before registration can take place, an initial guess must be provided. The currently used method is to pick several points/landmarks, and use their positions as a first approximation. There is a program called QtVisualizationTool (vis) that takes care of most of the process. The images are first read into memory. In this specific case, these are fetal mice, that belong to the same strain and at the same stage in development. In order to more easily distinguish between the fixed and moving image, the fixed image has been coloured red, and the moving image has been coloured green. During the point picking process, it is recommended that the image not being landmarked be made invisible, to avoid any errors.

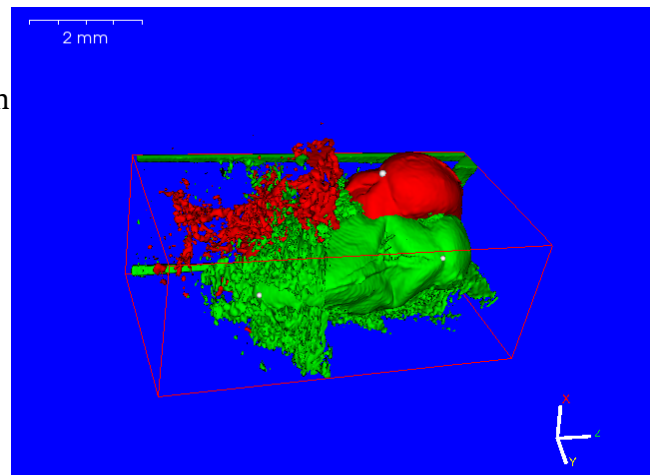


Illustration 3: Step 1: read in the images

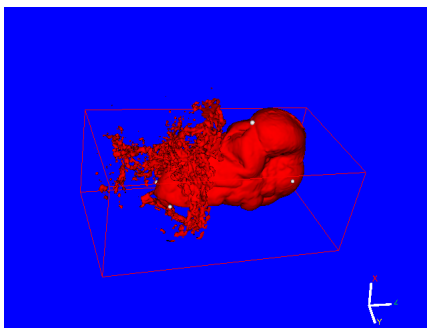


Illustration 4: Step 2: pick points on the fixed image

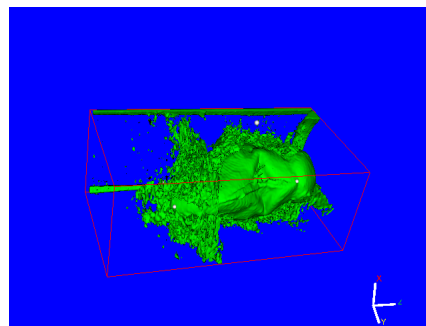


Illustration 5: Step 3: pick points on the moving image

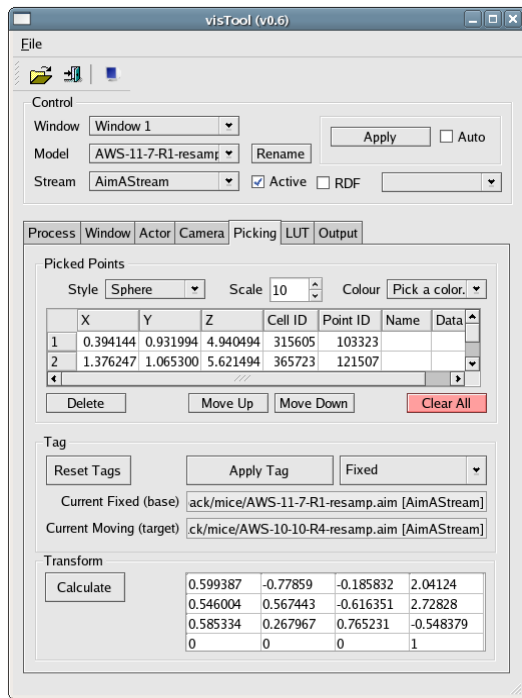


Illustration 6: Step 4: calculate the transformation

The program does not actually return the Euler angles, but rather the sin of those angles. This was a design decision by the authors of ITK. These values can then be put into a configuration file that will be read by the registration program. The fields are as follows:

- debug – changing to 1 will display information about each iteration, but slow down the program.
- initial guess: the seven parameters – three rotations, three translations, and an isotropic scale factor.
- optimizer parameters – usually can be left alone. If a rigid registration is desired, the weighting for scale can be set to a large value, such as 10^{16} .
- metric parameters – if a faster registration is needed, the number of spatial samples can be decreased. This increase in speed comes with a decrease in accuracy.
- other parameters – these change for each dataset and require some experimentation to set correctly. Having more than 3-4 levels causes the first level to be of too low resolution, and the solution to potentially diverge. The number of iterations per level, and the minimum change, are determined by how much time is available.

The registration can now be run. This specific example took approximately 25 minutes of computer time on an average desktop computer.

Once the points have been picked on both images, the program can be asked to calculate the best matrix that will align the corresponding points. Currently, the program is designed to find the best rigid transformation, but it still serves as a good first approximation.

The 4x4 matrix is a mathematical trick used to incorporate translation into matrices. The rotation, and potentially scale, is contained within the first three rows and first three columns. The first three rows of the last column represent translation.

The similarity 3d transform uses this 4x4 matrix internally, but this matrix must be decomposed to the three Euler angles. A converter program was written to do this. Eventually, it may be incorporated as part of vis.

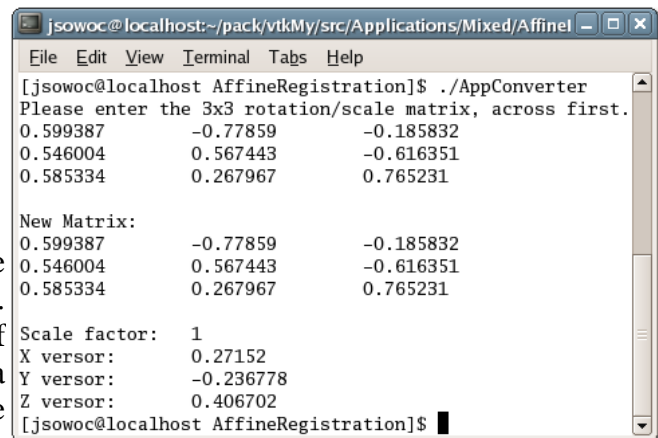


Illustration 7: Step 5: convert to Euler angles

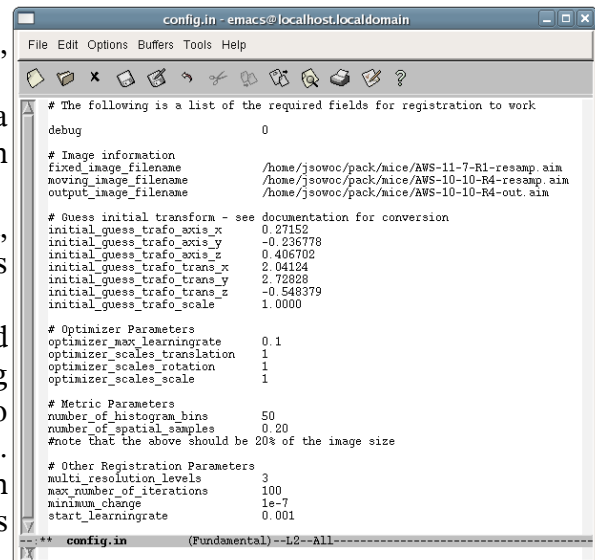


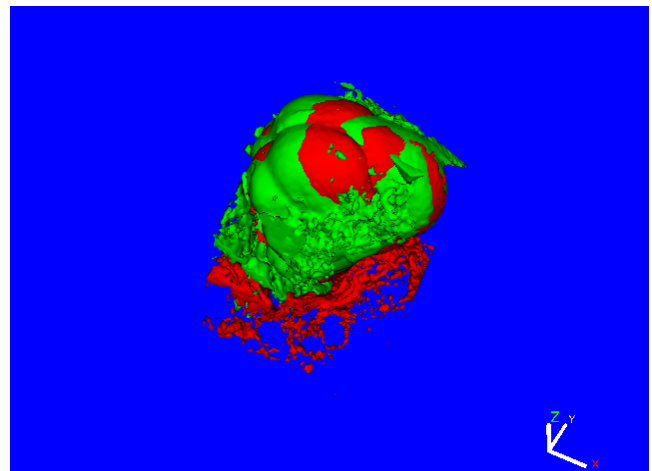
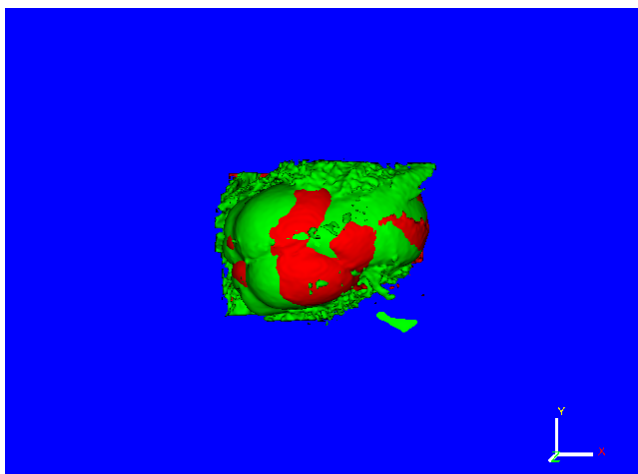
Illustration 8: Step 6: put the data into a configuration file

```
jsowoc@localhost:~/pack/vtkMy/src/Applications/Mixed/AffineRegistration
File Edit View Terminal Tabs Help
[jsowoc@localhost AffineRegistration]$ ./AppAffineRegistration -config config.in

0 = 0 : []
Scale was set to 64
Iteration number: 10
Iteration number: 20
Iteration number: 30
Iteration number: 40
Iteration number: 50
Iteration number: 60
Iteration number: 70
Iteration number: 80
Iteration number: 90
Iteration number: 100
100 = -0.356686 : [0.260149, -0.213707, 0.399036, 2.04567, 2.72699, -0.542362, 1.01081]
Scale was set to 8
Iteration number: 110
Iteration number: 120
Iteration number: 130
Iteration number: 140
Iteration number: 150
Iteration number: 160
Iteration number: 170
Iteration number: 180
Iteration number: 190
Iteration number: 200
100 = -0.2844 : [0.260222, -0.21479, 0.392007, 2.04988, 2.72286, -0.535007, 1.01138]
Scale was set to 1
Iteration number: 210
Iteration number: 220
Iteration number: 230
Iteration number: 240
Iteration number: 250
Iteration number: 260
Iteration number: 270
Iteration number: 280
Iteration number: 290
Iteration number: 300
Result =
Transform parameter 0 = 0.259768
Transform parameter 1 = -0.216501
Transform parameter 2 = 0.386575
Transform parameter 3 = 2.05699
Transform parameter 4 = 2.71694
Transform parameter 5 = -0.522598
Transform parameter 6 = 1.01048
Iterations = 100
Metric value = -0.257654

! Image data starts at byte offset 0
!-----
Volume /home/jsowoc/pack/mice/AWS-10-10-R4-out.aim
AIM Version 1.6
!
dim 253 186 500
off 0 0 0
pos 0 0 0
element size in mm 0.0125 0.0125 0.0125
phys dim in mm 3.1625 2.3250 6.2500
!
Type of data short (16 bit)
!-----
[jsowoc@localhost AffineRegistration]$
```

Following three hundred iterations, one hundred at each of three levels, the program finished. The final transformation was close to the original guess.



The fact that the two mouse images are almost perfectly superimposed shows that registration worked. The fixed and moving images weave in and out of each other.

Actual data that would have used the scale factor was not available in time for publication. To verify that scale factor does work, tests were run on artificial data. A z-bar was registered to another z-bar of a different size. The registration found the correct size difference to within a few percent.

Discussion

In order to develop a finished product, several design decisions had to be made. The decision to use a similarity transform and a linear interpolator was fully warranted based on previous research. The input file type is confined to an AIM file, as the majority of files to be registered in the lab are AIM files or can be converted to AIM files.

The choice of optimizer was not as trivial. The Gradient Descent Optimizer was selected because previous research [5] found that it worked. This optimizer was compared to the Regular Step Gradient Descent Optimizer, and was found to be superior. There are about a half-dozen other optimizers in ITK that may or may not be better.

The choice of metric was not as straightforward, and may turn out to be incorrect. The choice to go with Mattes Mutual Information was based solely on the experience of others with micro-CT data [5]. Previous experience has shown that Mattes works, and the solutions are satisfactory. ITK gives the user a choice to use any of about a dozen different metrics.

Given 12 possible metrics and 10 possible optimizers, it would be a project of its own to gather a large set of representative data to be registered. The registration would have to run several times on each combination of metric and optimizer. It was beyond the scope of this research to investigate further. A hindrance to this future research would be the subjectiveness of the accuracy of registration. It was observed that the metrics and optimizer behaved differently on “real” data as opposed to data that was generated and transformed a known amount.

In the process of creating the image pyramid, it was decided that the bottommost level should run at full resolution, which was not necessarily the case with the original rigid registration program. This change was justified in two ways. The first is that the input images can be scaled down before running registration. Micro-CT images are often quite large, and if the computing power is not available to register them at full resolution, the images should be pre-shrunk. In the case of the trial run, the images were pre-shrunk twofold in each direction. If this was not done, the registration would run for approximately four hours. The second justification for running the last run at full resolution, is the fact that the number of spatial samples can be easily adjusted in the configuration file. If this number is decreased, the optimizer does not need as many samples and the interpolator has less work to do. Changing the number of spatial samples from 0.20 to 0.020 reduced the computing time from twenty five minutes to three minutes. This increase in speed came at the cost of a less accurate registration. This reduction of spatial samples is equivalent to running the registration at a lower resolution.

Because this registration program uses a similarity transform, it is able to incorporate isotropic scaling. The creators of ITK realized that weighting factors needed to be assigned to the parameters, as a change of 0.1 in terms of rotation is not the same as a change of 0.1 in terms of translation. Using this feature, it is possible to “disable” scaling if a rigid registration is desired. If one were to set the optimizer scales for scale to a high number, the optimizer would not change the scale much. Set sufficiently large, the scale would not change at all, and the program reduces in function to the original rigid registration program.

Constructing a New Framework

For the last few years, all registration in the lab was conducted using the original Multi-Resolution Registration program. Over the years, many auxiliary programs have been written to visualize, pick points, and transform AIM files. Because the Multi-Resolution Registration program worked with the rigid quaternion transform, most of the auxiliary programs work only with quaternions. Over time, this software needs to be rewritten to work with the similarity transform.

The most important program that needs to be rewritten is the point picker within vis. This tool finds the transformation that best maps the points from one image to the other; it is coded to use a rigid transform, and the matrix it returns is a rigid matrix. A helper program (AppConverter) was written that converts a 3x3 matrix into scale and rotation, but the matrix returned by vis does not incorporate scale. Although this is a relatively simple correction, vis is used for many other routine tasks and should not be altered lightly. AppConverter correctly extracts (using SVD) the angles, but the user still needs to guess blindly at the scale factor.

The purpose of registration is to find the transformation that best maps the moving image to the fixed image. This is not useful by itself; the moving image needs to undergo the transformation in order to be useful. The registration program will automatically output the moving image transformed to the size and shape of the fixed image. This is not sufficient because, as mentioned earlier, a downsampled version of the image is often registered. A program named AIMTransformer is then able to transform the original image by the appropriate amount. This program does not yet natively support the Euler angles/scale factor. This needs to be rectified in the near future.

Conclusions

A program was written that implements a similarity transform, and works on micro-CT data. It uses a Mattes Mutual Information Metric, a linear interpolator and the Gradient Descent Optimizer. This combination of components was chosen based on previous research. Because the use of a scale factor can be effectively disabled, the new program is a complete replacement for the quaternion-based rigid registration program.

References

- [1] T. S. Yoo, "Chapter 10: Medical Image Registration: Concepts and Implementation," *Insight into Images*. Wellesey, MA: A K Peters, Ltd.; 2004
- [2] J. M. Fitzpatrick, D. L. G. Hill, C. R. Mauer. "Chapter 8: Image Registration," *Handbook of Medical Imaging*, Volume 2: Medical Image Processing and Analysis. Bellingham, WA: SPIE Press; 2000, pp. 447-513
- [3] L. Ibanez, W. Schroeder, *ITK Software Guide* (chapter 8), <http://www.itk.org/ItkSoftwareGuide.pdf>
- [4] J. Hajna, D. Hill, D. Hawkes, "Chapter 2: Registration Methodology: Introduction; Chapter 3: Registration Methodology: Concepts and Algorithms," *Medical Image Registration*. CRC Press; 2001
- [5] M. Kuhn, S. Moser, *Development of Intra-Modality Image Registration Methods for Assessment of Micro-Structural Bone Adaptation*, Semester Project, Summer 2003
- [6] J. B. Kuipers, *Quaternions and Rotation Sequences*. Princeton NJ: Princeton University Press; 1999