Practical, Computer-Aided Registration of Multiple, Three-Dimensional, Magnetic-Resonance Observations of the Human Brain

Carl Diegert[1], John A. Sanders[2], and William W. Orrison, Jr[3].

RECEIVED
JUL 2 6 1993
OSTI

1. Introduction

We define a methodology for aligning multiple, three-dimensional, magnetic-resonance (MR) observations of the human brain over six degrees of freedom. The observations may be taken with disparate resolutions, pulse sequences, and orientations. The alignment method is a practical combination of off-line and interactive computation. An off-line computation first automatically performs a robust surface extraction from each observation. Second, an operator executes interactively on a graphics workstation to produce the alignment. For our experiments, we were able to complete both alignment tasks interactively, due to the quick execution of our implementation of the off-line computation on a highly-parallel supercomputer. To assess accuracy of an alignment, we also propose a consistency measure.

Researchers working with MR observations generally agree that far more information is available in a volume (3D) observation than is considered for diagnosis. The key to the new alignment method is in basing it on available information on surfaces. Using the skin surface is effective because: (1) a robust algorithm (like the one we define) can reliably extract this surface from almost any scan of the head, and (2) a human operator's exquisite sensitivity to facial features is allows him to manually align skin surfaces with precision.

Following the definitions, we report on a preliminary experiment where we align three MR observations taken during a single MR examination, each weighting arterial, venous, and tissue features. When accurately aligned, a neurosurgeon can use these features as anatomical landmarks for planning and executing interventional procedures.

The need for practical, accurate alignments is broad, extending beyond the specific experiment we describe. For example, alignments are pivotal in studying aging and disease progression from observations taken over time. Aligning observations, all taken of the same patient, but taken by different modalities, can help bring precision and efficiency to diagnoses now taxing the time and skill of radiologists. Volume acquisitions by MR, CAT, and PET are already commonly taken in various combinations. Non-subjective products from an alignment, such as computer-aided control of stereotactic fixtures, are possible only when multi-modality data are aligned and fused into a single, quantitative dataset.

2. Alignment Method

This paper defines an accurate, computer-aided method to align multiple, magnetic-resonance observations of a patient's head. The particular method we define is a snapshot of our evolving methods.

To align multiple scans of a patient's head, we settled on finding, then visually aligning the patient's skin. Aligning skin, in preference to bone structure or some other feature, offers two advantages: first, most magnetic-resonance observations of the head include information on the location of the skin, as there is a sharp transition from no signal outside the head, to at least some signal starting at the skin boundary. In fact, skin and subcutaneous fat typically produce a strong signal. Second, human operators are extraordinarily sensitive to facial features, and can therefore accurately align skin surfaces.

For the alignment method, we found simple algorithms that work effectively together: first we apply off-line filtering and surface extraction steps, then we execute interactively to produce a computer-aided alignment. In addition to their

simplicity, the off-line steps can execute well as local operations on distributed-memory, parallel computers. After filtering and surface extraction, our method relies on a human operator to interactively align the observations, guided by a computer's rendering of these surfaces. Again we chose simplicity, providing the operator with depth-cued, wireframe renderings of the extracted surfaces.

Only the last step of the alignment method, aligning the depth-cued, wireframe renderings, must execute interactively. This step executes effectively and economically using off-the-shelf software running on an inexpensive graphics workstation. For the experiments described here, we used a Silicon Graphics workstation equipped with their VGX hardware to accelerate the rendering. Interactive manipulation of large, wireframe objects is a part of dozens of software packages in use at Sandia National Laboratories. For these experiments, we used a commercial geometry viewer package (Advanced Visual Systems, Waltham, MA).

While filtering and surface extraction might run non-interactively, on large-memory workstations, we accelerated our development and experimentation by running the whole alignment method, including these steps, interactively. To run the compute and memory-intensive steps interactively, we hand-crafted an implementation of these steps for distributed-memory, MIMD computers, and executed them interactively on one of Sandia's 1024-processor, distributed-memory nCUBE-2 supercomputers (nCUBE Corporation, Foster City, CA).

2.1 Filtering

For the alignment method we succeeded by combining simple filtering, as defined next, with a marching-cubes[1] surface extraction. Without departing considerably from the marching-cubes algorithm and its widely-used variants, some filtering of magnetic-resonance observations is essential. Marching-cubes surface extraction is sensitive to noise, in general too sensitive for direct processing of MR observations. Our future direction will be to attain better precision by extracting surfaces in one step, using statistical estimators to properly and efficiently directly treat noisy observations. In this case, no separate filtering step would be applied.

A simple thresholded, trimmed-mean filter proved to be effective, and we used it consistently to produce the results reported here. The filter is applied locally and independently to each voxel. If the voxel's value is less than a threshold we leave the value unchanged. We used a threshold value just above most of the low, noise signal observed outside the head. If the voxel's value exceeds the threshold, a minimum and maximum of the values of the six neighbor voxels are

dropped, to yield a list of four values. The new value for the voxel is a weighted sum of the old voxel value (with self-weight 1/3) and the mean of the four remaining adjacent values.

We found that applying two passes of the filter produces easily-recognizable features in the skin surface, and recommend this procedure for off-line filtering implementation. More filtering certainly can remove more noise, but may also introduce bias in the estimated location of the skin.

For our experiment, filtering, surface extraction, and surface rendering all executed quickly on a distributed-memory supercomputer. Using this compute power, we departed from the off-line recipe by interactively choosing an appropriate amount of filtering for each MR observation. In each iteration of this interaction, we applied a trial filter and viewed the resulting rendered surface few seconds later. For example, analyzing the 256x256x128 voxel scan (observation T, described in the next section) on 512 nodes of our nCUBE-2 supercomputer, a pass of trimmed-mean filtering completed in 1.06 seconds. Extracting a surface from the filtered data produced 795,166 triangles (without the structural modification described below), and completed in 1.31 seconds. With the structural modification, the code produced 343,397 triangles, also in 1.31 seconds, after using about two seconds to execute the structural modification.

We obtained all communication and synchronization for our parallel implementation of the filter with calls to a portable library written by Mark Sears[2]. This implementation distributes voxel data spatially across a parallel computer's processors, with each processor receiving a spatially-contiguous block of voxel values. For the local operations of the thresholded, trimmed-mean filter, a processor needs only a few values stored on other processors to filter voxels at the edge of its voxel block. We distribute the voxel data, and arrange for these references using the library's support for repeated, or ghost, voxel values.

2.2 Surface Extraction

After filtering MR observations, we obtained surfaces using a simple marching-cubes surface extraction algorithm[1], modified to include a structural computation. Our structural computation prevents generation of surfaces interior to the head.

For parallel implementation, the same spatial decomposition of voxel values used for our filtering step supports the marching-cubes computation. Our implementation remains cleanly layered on the processor communication and synchronization provided by calling routines from Mark Sears' library. Because filtering is centered on voxel values, while

Figure 1: Two surfaces, both extracted from MR observation T. Left used unmodified extraction, right used a structure modification.

the marching-cubes algorithm is centered on a cube formed by eight, mutually-adjacent voxel values, we used an additional layer of repeated, or ghost values to allow execution of both filtering and marching-cubes surface extraction from the same distributed data.

In addition to working with us to extend his library, and to apply it to making MR alignments, Mark wrote and tested both a parallel rendering code, and a parallel implementation of marching cubes. We incorporated both in the code for our alignment work. Mark's implementation of marching-cubes for parallel execution is an adaptation of a serial code by Mike Krogh. [Mike's code, isovis, is public-domain software from National Center for Supercomputer Applications, Univ. Illinois, Urbana-Chapaign, IL]

Without modification, the filter followed by marching-cubes surface extraction can extract the skin surface from most MR observations. However, these steps produce a great many surfaces interior to the head together with the skin surface. When rendered as surfaces, as we do in our supercomputer implementation, the skin surface must be cut away to reveal the interior clutter. We clipped the heads in Figure 1 to show some of the interior surfaces. Surfaces that comprise the head on the left, a total of 795,166 triangles, are from our unmodified, marching-cubes implementation.

We found that, with the simple, depth-cued wireframe rendering used to provide feedback in the final, interactive set of the alignment, the clutter of interior surfaces distracted the operator and greatly complicated his interaction with the

computer display. To eliminate interior surfaces, we added code to generate a mask from the voxel data, and modified our marching-cubes implementation to generate surface triangles only within voxel cells selected in the mask. For these results, we used a structural computation to generate the mask: we chose the coordinate axis closest to the ear-to-ear axis, and dropped out any voxel cells between the first and last cells (along this axis) that contributed to the surface. For one of the observations we analyzed, the acquisition did not extend to the ears. For this observation, we chose the axis extending anterior to posterior.

The right head in Figure 1 shows a surface like that on the left, but produced under the structural modification. The right head surfaces comprise 343,397 triangles. The simplicity of the structural computation shows, for example, in that the first surfaces at the ears prevent (shadow) the skin behind the ear from contributing to the picture.

Our parallel implementation of the structure computation followed the data decomposition, communication, and synchronization used for filtering and for marching cubes extraction. The computation first finds an array of label values, labeling only voxels at of beyond the first voxel to contribute to the surface, as one proceeds along the ear-to-ear axis toward higher-valued coordinates. This labeling proceeds in phases. At the first phase, each processor completes the voxel labeling as though its sub-block of voxels comprised the entire observation by labeling first and subsequent voxels from the block of voxels stored locally in its memory.

For this phase all voxel values, including ghost values, are processed. Thus, this phase produces a spatially-decomposed mask array, complete with ghost mask values. The second phase is to synchronize and update the label array ghost values. This update propagates the determination of what is locally first along the chosen axis. Iterating the two phases until the update does not change any label values obtains the globally-correct labels. The number of iterations is, at most, one less than the number of processors in the decomposition of a global line of label values along the chosen axis. Next we find second array of label values, labeling from the first contributing voxel as one proceeds along the ear-to-ear axis in the other direction. Finally we combine the resulting masks with an element-wise, boolean AND operation. Implemented in this way, the structure computation executes quickly on our distributed-memory computer, and uses the same library calls that support the filtering and surface-extraction computations.

2.3 Terracing

The layers, or terraces, in the skin surfaces shown in Figure 1 are an artifact of the quantization of voxel values on the MR instrument. In our acquisitions, signal values of about 15 defined the transition to skin from a near-zero noise signal just outside the head. Compounding the quantization artifacts are small ringing artifacts, likely produced by frequency-limited response of the MR instrument to the sharp transition from little signal outside the head to the strong signal returned by skin and subcutaneous fat. We found that an operator could easily ignore the terracing in the final, interactive step. Figure 2 shows a surface from an observation of
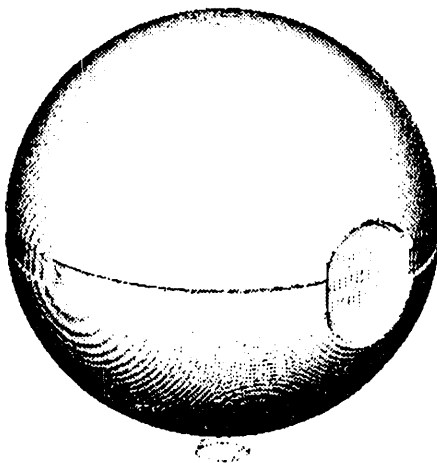

Figure 2: Surface extracted from an MR observation of a smooth phantom.

a smooth phantom, and confirms that ringing occurs in all coordinate directions, and along combinations of coordinate directions.

2.4 Interactive Alignment

Commercial animation and analysis software packages, executing on graphics workstations, offer an operator fluid and refined interaction with geometric objects of substantial size. For each pair of the three MR observations described in the next section, we used the geometry viewer module from Advanced Visualization Systems (AVS), Waltham, MA, to rotate and translate one surface to best match the other. To allow interaction with the largest-available surfaces, we used AVS library routines to include only wireframe information in the AVS geometry object, and to organize these objects for efficient display (as polytriangle objects). Despite these provisions, we could not effectively interact with the full-resolution surfaces, each with about a million triangles, when executing on our Silicon Graphics model 4D/340VGX workstation. To complete the interactive alignment, we added a downsampling step between our first two steps. After filtering the MR observation at full resolution, we downsampled the filtered data, using only every other voxel value along each coordinate direction, thus retaining one-eighth of the voxel values. We extracted surfaces from this smaller, subsampled dataset.

To provide feedback for our alignment of surfaces we used the geometry viewer to define multiple, simultaneous views of the surfaces (multiple cameras), and to manipulated these views in order to zoom in on feature details in the face, the ears, and elsewhere. Single views of the surfaces, even on a mega-pixel monitor, were not sufficient to guide our interaction. We worked with color and depth cueing applied to all views.

3. Application to Neurosurgery

In this section we report on aligning three disparate, MR observations of the head of a normal volunteer. The combined data contain arterial, venous, and other tissue structures. In a diseased patient, lesions would appear together with the vascular structures. Once registered, we used the combined data to generate an accurate network of anatomical landmarks, useful to the neurosurgeon for planning and executing a difficult intervention.[3]
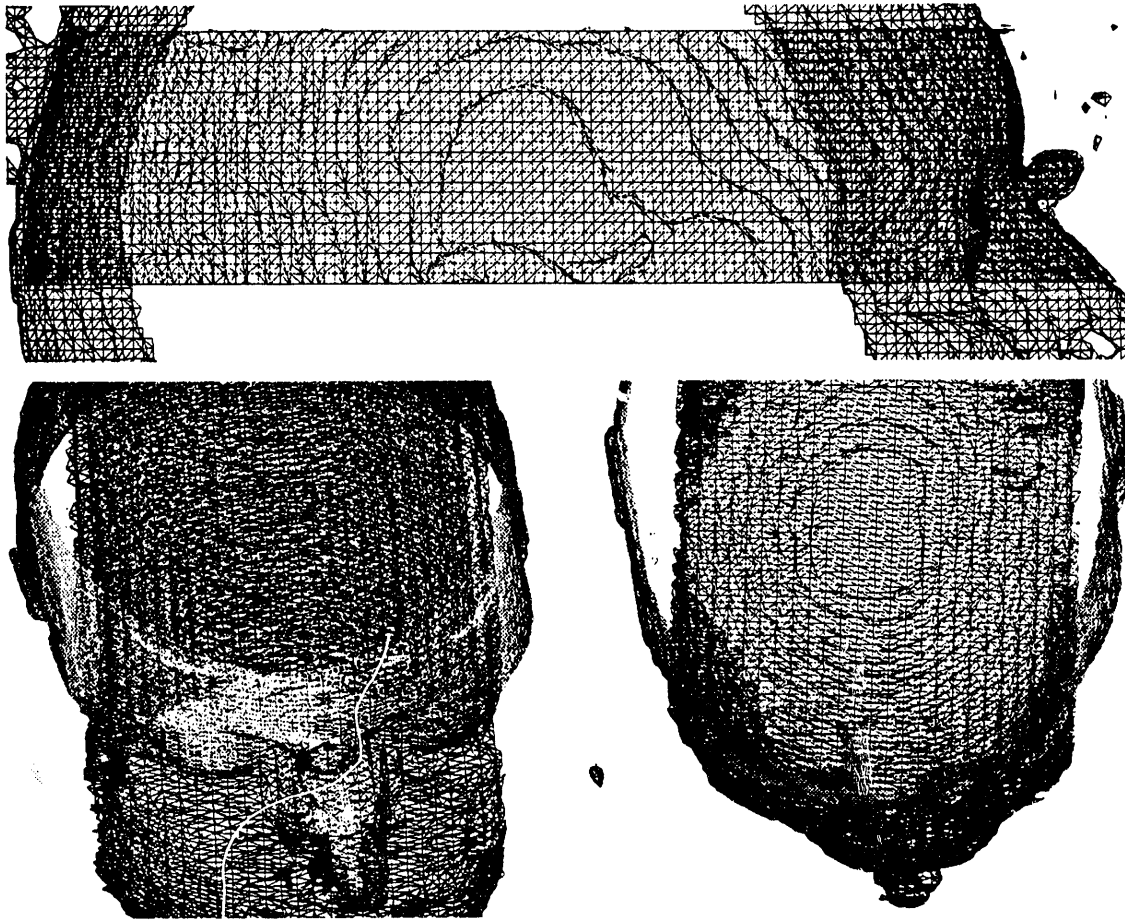
Figure 3: A display for visually checking the consistency of three pairwise alignments of observations T, A, and V. This figure shows observations A and V, as aligned with reference to observation T, but not with reference to each other.

Table 1: Parameter values for three MR observations.

Code	Observation	Acquisition Time (minutes)	Pulse Sequence (TR,TE,Angle)	Slice Thickness, Spacing (mm)	Number of Slices
T	High-Contrast, Anatomic Data	9	MPRAGE (10,4,10)	1.5, 1.5	128
V	Slow Flow in Venous Vessels	8	FLASH2D (32,10,50)	5.0, 2.5	54
A	Rapid Flow in Arteries	8	FISP3D (29,7,50)	1.0, 1.0	64

3.1 The MR Observations

In a single MR examination, we acquired datasets from three observations of a normal volunteer on our clinical, 1.5T imaging system (Magnetom, Siemens Medical Systems, Inc., Iselin NJ). We call these observations T, V, and A. All acquisitions imaged over a three-dimensional volume, with an array of 256 by 256 voxel values recorded for each slice plane. All slices have 1.0 mm, in-plane resolution. Parameter values for T, V, and A appear in the Table 1. For observation V, we improved the time-of-flight measurement of the relatively slow, venous flow by increasing the slice thickness, and by overlapping the sequentially-acquired slices by fifty percent. Rotating the acquisition by about 10 degrees from sagittal also improves this observation.

3.2 Filtering and Surface Extraction

We executed filtering, followed by surface extraction on our highly-parallel supercomputer. Given the speed of this iteration, just a few seconds, we improved slightly on our suggested off-line prescription for blindly applying two filtering passes. With interactive execution on the supercomputer, we displayed the surface after each filtering pass using a quick rendering algorithm, also executing on the supercomputer. We then examined the surface, looking for a clean presentation of facial and other details. If noise obscured these details, we applied another filtering pass. The result was to apply a single filtering pass to observation T, and two passes each to observations V and A.

The next step on our supercomputer was to downsample the filtered data, retaining every other voxel along each coordinate direction. The last supercomputer step, then, was to extract a surface from the downsampled data, and to dump it to a file for transfer to a graphics workstation. Table 2 lists

Table 2: Surfaces for alignment

observation	filtering passes	surface triangles from all data	surface triangles from downsampled data
T	1	343,397	75,793
V	2	234,831	52,232
A	2	108,468	26,188

the sizes of the resulting surfaces.

3.3 Alignment Results

For this preliminary experiment, a single operator completed a single set of the three possible pairwise alignments of the three MR observations T, V, and A. All three observations produced excellent skin surfaces, but the coverage varied, and determined the pairing we used. Because coverage of the angiography observations, observations V and A, was limited, we registered each of them to the full-coverage observation T. As a check on the accuracy of the alignment, we also performed a third alignment, aligning observation A to V. The full coverage of observation T is suggested by its rendering in Figure 1. The limited coverage of observations A and V is apparent from Figure 3. Observation A covers all areas in axial slices through the head, and extends upward from the top of the ears for 6.4 cm. Observation V covers all areas in sagittal slices of the head, with 13.4 cm coverage along the ear-to-ear axis, approximately centered on the head. Figure 3 shows that observation V does not extend out to either ear.

For each alignment, we loaded the two wireframes into the geometry viewer, and assigned each of them a unique color. Rather than modify the geometry viewer software, we just carefully avoided any interaction that would alter the scale of either wireframe, or alter any transformation parameters of the base wireframe. This left us free to alter translation and rotation of the non-base wireframe, and to alter all parameters of multiple cameras on these objects. While no lighting applies to these wireframe renderings, we did make use of altering the depth-cue parameters of the cameras, as well as the color of the wireframe objects. With no time limit constraint, we stopped each alignment after about 10 minutes of interaction, when we judged that further work would not improve our result. At the end of each alignment, we used the geometry viewer's command-line interface to record the alignment result as a four-by-four transformation matrix.

Our only measure of accuracy in this preliminary experiment is assessing consistency of the three alignments. The greater number of surface details in the V-T and A-T pairings made these alignments easier, and likely more accurate than the A-V alignment. As a visual check of this accuracy, we applied the transformation matrices from V-T and A-T to the V and A wireframes. Figure 3 shows views of the resulting V and A objects. With perfect alignment, V and A would be correctly registered.

The result of our third alignment, A-V, allows us to make a quantitative consistency check. For this check, we computed a T-A transformation by numerically inverting the A-T

alignment transformation. We then numerically composed the three transformations T-A, A-V, and V-T. Inconsistencies in our three alignments cause departures of the composed transformation from an identity transformation. Table 3

Table 3: Errors when composite transformation matrix from three pairwise alignments maps coordinates from observation T back to T. Coordinates and errors are in millimeters.

Location	Location Coordinate	Length of Error Vector
Nose tip	175.9,-154.4,4.0	1.21
Head back	90.0, -138.7, -229.7	1.88
Head top	85.5, -39.4, -107.9	0.88
Right Ear	9.4, -170.0, -130.0	1.42
Left Ear	174.0, -162.7, -130.0	1.38

gives the error at four points. Among these locations, the greatest error on this single trial is 1.88 mm, and occurs at the back of the head, where skin features are less prominent than at the face and ears.

4. Conclusions

Indentifying and manually aligning the skin surface from multiple MR observations of the head is key to obtaining accurate alignment of these data. This paper defines a specific method, and reports on a preliminary experiment obtaining six-degree-of-freedom alignments. As a combination of off-line and interactive computation, the alignment method is practical, executing on commonly-available computer hardware. As highly-parallel computers become available our parallel implementation will become more practical.

Before beginning a more comprehensive experiment, we plan to implement a more effective final, interactive alignment step. We will soon receive a higher-performance workstation, capable of smooth interaction of the surfaces we extracted without first downsampling the data. We will also move to using new software in place of the commercial geometry viewer for the interactive alignment. The commercial geometry viewer in its choice of functionality, and in how its interface to the operator was defined and implemented, complicated the manual alignment. We may be able to obtain better alignments using a commercial package

designed for animation, rather than one designed for scientific data analysis.

With the more comprehensive experiment, we will identify sources of errors in the alignment, including variability from operator to operator, and how variability grows when downsampling lowers the surface resolution. Sensitivity to terracing artifacts could be addressed. Additional experiments will be the basis for tuning the alignment method, and then for assessing its accuracy. With these experiments we will be in a position to measure the benefit of replacing the filter-then-extract steps with a computationally-intensive, single-step surface estimation algorithm. We plan to implement and test several of these alternative estimation algorithms.

Statistical understanding built on these experiments can help guide development of a more fully automatic alignment method. The one-step surface extraction is a step in this direction, replacing a minor human intervention that determines when to stop filtering. A next step will be to let the human supervise the computer-aided alignment at higher levels. In a future implementation, a human operator would interactively define, and perhaps delete, pairs of features, one from each surface. The computer would then automatically align features by automatic optimization of correlation-like statistics. The operator's high-level understanding of facial and other features will bound the size of the automatic computation, by limiting the domain of the optimization. The operator will also make the procedure robust, by recognizing and correcting gross errors, errors not apparent in the correlation statistics.

5. References

1. W. E. Lorensen and H. E. Cline, "Marching Cubes: A High Resolution Surface Construction Algorithm," *Computer Graphics*, Vol. 21(3), pp. 163-169, July, 1987.

2. M. P. Sears, MPSLIB - "A Portable, Massively-Parallel Software Library", *Sandia National Laboratories Tech. Report SAND-930741*, June, 1993.

3. W. W. Orrison, Jr., J. D. Lewine, J. A. Sanders, J. George, C. Diegert, and R. D. Bucholz, "Functional Brain Imaging," *Hot Topics Presentation*, Radiological Society of North America Meetings, Chicago, December 1992.

DISCLAIMER

END

DATE FILMED
9 / 15 / 93