SOFTWARE DEVELOPMENT FOR CORRECTION OF GRADIENT-NONLINEARITY DISTORTIONS IN MR IMAGES.

Tom S. Lee†, Keith E. Schubert*,*

California State University, San Bernardino
Department of Computer Science
San Bernardino, CA, USA

Reinhard W. Schulte†,

Loma Linda University Medical Center
Department of Radiation Medicine
Loma Linda, CA, USA

ABSTRACT

A new system for functional proton radiosurgery has been proposed. The goal of the system is to target specific brain areas with high doses of proton beams with submillimeter accuracy. High-energy proton beams have exquisitely sharp lateral penumbra and are, therefore, ideal for functional radiosurgery. Localizing the anatomical target with an MRI-based fiducial system requires correction of gradient nonlinearity distortions inherent in the scanner images. Modern MR scanners are particularly prone to such distortions due to wider bores and stronger gradient fields. The gradient nonlinearity correction described in this work is based on a high-resolution 3D MR scan of a cube phantom. Using a least-squares fitting procedure correction parameters are found that convert the geometrically warped planes of the cube into the ideal planes. In this paper, we describe the initial data processing and quality checks performed before the data is used for estimation of correction parameters.

1. INTRODUCTION

Functional radiosurgery is a non-invasive treatment technique for creating small lesions in diseased areas of the brain to treat symptoms of functional disorders such as trigeminal neuralgia and Parkinson's disease [1]. A system for functional proton radiosurgery based on the sharp penumbra of high-energy proton beams (250 MeV) is currently being developed at Loma Linda University Medical Center. Successful application of this method requires a high degree of geometric accuracy in localizing the intracranial target with high-resolution MRI. Nonlinearity of the magnetic gradient fields, the most prominent source of geometric distortion in modern MR scanners, can negatively influence the accuracy of target localization [2].

Several MRI distortion correction methods have been described in the literature, which are usually based on scanning a phantom with accurately known reference structures [3, 4, 5, 6]. These are detected by a specifically designed software, and a distortion correction model for the distortion between the ideal and the detected structures is assumed for which the model parameters are estimated, usually by a least-squares fitting procedure.

In this paper, we present in detail the initial steps in the software development that will correct for gradient nonlinearity distortions. Our software is based on a cubic phantom scan as previously described [6]. We will discuss challenges encountered in the software development process and will describe how they were solved.

*Supported by the NSF under award CISE 9810708.
†Supported by the Henry L. Guenther Foundation.

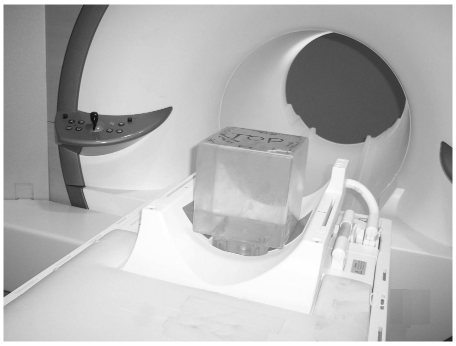


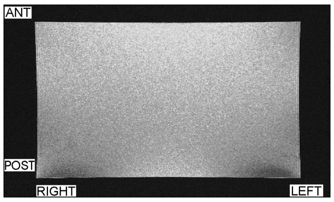
Fig. 1. Cubic phantom in MRI scanner

2. METHODS AND RESULTS

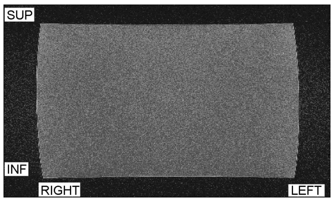
The gradient-nonlinearity distortion correction method of Langlois et al. [6] utilizes the MR scan of a cubic phantom filled with oil, shown in Figure 1. The signal response seen in the MR images represents the interior dimensions of the phantom (159.50 mm x 159.70 mm x 158.11 mm). MR images of the phantom were acquired on a 1.5 T whole body scanner (Magnetom VISION, Siemens, Germany) with a standard head coil. The phantom was centered on the gradient isocenter of the scanner to within 1 mm. To maximize the amount of available data for correction, axial, coronal, and sagittal 3D MPR sequences were acquired. The acquisition parameters were as follows: TR 2010 msec, TE 2.75 msec, flip angle 20 msec. A matrix of 512 x 512 pixels and a field of view of 200 mm were chosen. One slab of 104 slices of 2.0 mm thickness was acquired for each sequence. Thus, the voxel size was 0.391 mm x 0.391 mm x 2.000 mm. Figure 2 shows representative axial, coronal, and sagittal images of the phantom, with distortion clearly visible in the outer regions.

Prior to calculating the parameters of the distortion correction model, the data undergoes a series of image processing and quality checks described in detail below. The quality checks have three main goals: (1) eliminate as much image noise as possible, (2) remove extraneous and unwanted features in each MR image, and (3) ensure the data is as accurate as possible for future calculations. The ultimate accuracy of the correction will depend on the quality of the data produced after these data processing steps.

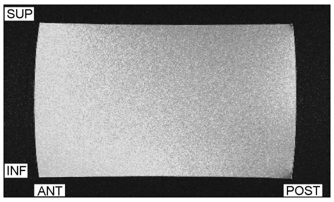
In the following we will describe the steps of phantom edge detection and quality assurance. Each quality check is executed as an individual module during the initial data processing stage. The software package performing these steps was developed with Matlab version 7.0.4.365 Service Pack 2.



(a) Axial scan of phantom.



(b) Coronal scan of phantom.



(c) Sagittal scan of phantom.

Fig. 2. Examples of useful slices from various MRI scans. Note the different scales of the horizontal and vertical axes.

2.1. SELECTION OF USEFUL SLICES

The distortion correction should only be based on slices that contain a full view of the phantom. To select useful slices, our algorithm analyzes the pixel intensities of each slice and compares them to a reference slice near the center of the slab, which will always contain a full view of the phantom. The central slice is not chosen as a reference because it contains the phantom's drain plug, which gives an inaccurate edge representation and intensity distribution (see Section 2.3).

As the first step in the selection of useful slices, edge detection (see next section) is performed on the reference slice, identifying the region that contains the phantom. Next, the maximum pixel intensity of the "empty" region surrounding the phantom (low-intensity value) and the minimum pixel intensity in the central 20% area of the reference slice (high-intensity value) are found. Two criteria are used to define useful slices: (1) the number of pixels with equal or larger than the high-intensity value has to be within 1% from the reference count, effectively eliminating slices without the phantom; (2) the relative number of pixels with intensities between the low- and high-intensity values has to be less than 40%, thereby excluding slices with a partial view of the phantom. The indexes of the first and last useful slices are reported, and slices outside this range are excluded from further analysis.

2.2. EDGE DETECTION

Edge detection is performed to provide the representation of the distorted physical edges of the phantom for the gradient nonlinearity distortion correction. The Canny edge detector (ED) provided by Matlab was chosen because it was least sensitive to noise, and the edge images produced were sharper and more accurate, thus delivering the most consistent results from study to study when compared to other methods implemented in Matlab.

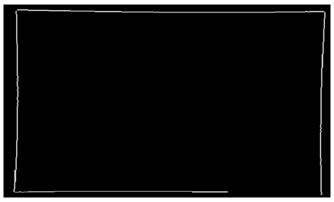
In the Canny ED, Matlab will automatically calculate a threshold for edge detection. However, this threshold considers every single intensity shift within the image, and in doing so, finds fictitious edges within the phantom. In order to find the optimum threshold value for our application, we have implemented a *threshold algorithm* that converges on the optimal value in an iterative series of edge detection trials. For the threshold algorithm, the reference image already selected for the definition of useful images is used. The threshold algorithm automatically configures its settings (threshold increment and stop condition) based on the strength of the image gradient across the edges. In the baseline iteration, Canny ED is performed using Matlab's automatically calculated threshold. Then, the threshold algorithm increments the threshold in small successive steps (0.5-0.75). As the threshold is increased, the number of fictitious edges decreases, leaving only the strong image features behind. The optimal threshold is assumed when the number of edge points does not change by more than 1% over several successive iterations (5-10, depending on the stop condition). This optimized threshold is then used with Matlab's Canny ED on the entire set of useful slices. In a series of tests, it was found that this will mostly retain the physical edges of the phantom in edge images. The remaining unwanted image features had to be dealt with separately, as described in the following sections.

2.3. DRAIN-PLUG REMOVAL

One image feature in the scan of our specific phantom that is not removed during edge detection is the phantom's drain plug, i.e., a screw that is located underneath the center of the posterior face. The threads of the screw are present in edge images acquired from axial and sagittal sequences. Tests showed that failure to remove the image of the plug will result in improper calculations in all subsequent data processing steps. Therefore, we implemented a *plug removal algorithm*.

In the first step of the drain-plug removal process, edge images are oriented such that the plug appears on the bottom face of the image. Removal of the plug is then performed by placing a window that has a width of 25% of the phantom width about the center column of each image. The plug removal algorithm removes all image data within this window, leaving behind a hole in the top and bottom phantom edges. Both holes are located by the algorithm and filled by a straight line. Since the deleted areas are located in the center of these faces, the distortion is small; at worst, the corrected portion of the edges is shifted by one pixel to either side of the edge. It was found that in all phantom scans, the plug removal algorithm properly connected the edge with a straight line, without introducing any gaps.

(a) Slice with oil leakage in left posterior corner.



(b) Detected edges of slice with oil leakage.

Fig. 3. Incomplete edge as a result of phantom oil leakage.

2.4. EDGE VERIFICATION AND REPAIR

The next step involves the detection of incomplete edge images, i.e., those with missing edge data. For example, in Figure 3, the phantom had leaked oil along the left posterior surface. The edge detection resulted in a large defect in the bottom right corner of several edge images.

In the edge-verification step, all edge images in an image sequence are checked for completeness. First, the location and size of the phantom is calculated by our *edge-verification algorithm*. Based on this information, the phantom edge image is split into four equal quadrants, and the total number of horizontal and vertical edge points in each quadrant is counted. In a complete edge image, the phantom will contain nearly the same number of edge points in each quadrant. On the other hand, in an incomplete edge image, the number of edge points in one quadrant will be different. These images, and the quadrant that appears anomalous, are flagged. Based on user preferences, problematic images can either be discarded from further consideration, or the images can be repaired. Image repair is accomplished by copying and grafting an intact quadrant from the opposite side in place of the flagged quadrant. It was found that the resulting image is very similar to that of a "natural" edge image, and is sufficient for use in further calculations. This repair worked successfully for all slices affected by the leaking oil, shown in Figure 3.

2.5. DETECTION AND REMOVAL OF AIR BUBBLES

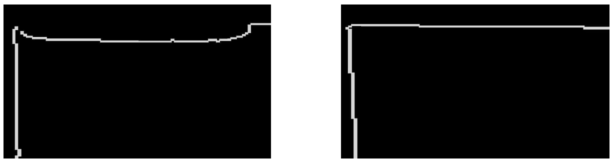
Due to the use of oil as a medium for the phantom, it is virtually impossible to remove all air bubbles. Small air bubbles that float within the confines of the phantom are not problematic, but those attached to the phantom surfaces can distort the phantom edges. Since the signal response seen in the MR images is that of the oil, air bubbles along the edges appear as concave defects of varying size. Air bubbles attached to the phantom corners give the false impression of a rounded corner. The study performed on the leaking phantom mentioned above resulted in a large air bubble in the right anterior corner of the phantom, as shown in Figure 4(a).

Air bubbles are detected in several steps. First, the locations of the four corners in each image are determined using our implementation of a *corner detection algorithm*. The algorithm defines a corner as the point shared by a horizontal and vertical edge (i.e., where those two edges meet). The corners are located by scanning up and down the vertical edges of the phantom (depending on which corner is to be located) until the start of the adjacent horizontal edge is found. When the horizontal edge is found, *and* there are no additional edge points found along the vertical edge, the true corner is considered located. This extra step is taken to avoid the algorithm getting confused by an air bubble along that edge, as air bubbles may resemble corners. By scanning beyond a possible corner location, the algorithm will disregard air bubbles along an edge as possible corners, and will only report the location of the true corner.

The corner data is then used by our *edge tracking algorithm*, which was implemented to detect air bubbles . Starting from one corner, it moves along the edge, pixel by pixel, recording the location of each edge point. Horizontal edges are scanned from left to right, and vertical edges are scanned from top to bottom. The edge tracker will stop once it has reached the opposite corner. In the next step, the edge *slope* is calculated for edge intervals of 5 points. Along a "normal" edge, the edge slope does not exceed one pixel over this range. On the other hand, an air bubble will cause a sharp positive (>1 pixel) rise in slope, which is followed by a similar drop on the opposite side of the bubble. The range between the rise and drop of the slope is considered as the limits of the bubble.

If the bubble is located on the edge rather than in a corner, it is erased, creating a hole in the edge, and the image is processed by our *hole repair algorithm*, described in the next section. On the other hand, air bubbles located in a corner are treated by replacing the entire corner with a copy of the opposite bottom corner, to avoid inserting another bubble in place of the original. The likelihood of having air bubbles in *both* bottom corners of one image was considered as *very* low. The bottom corners are tested for air bubbles first, and are corrected by the mirrored opposite bottom corner, if necessary. Next the top corners are tested for bubbles and replaced by the corresponding bottom corner, if air bubbles are detected.

Figure 4 demonstrates the results of removing the large air bubble present in the study with the leaking phantom.



(a) Before correction. (b) After correction.

Fig. 4. Demonstration of bubble correction on large corner bubble.

2.6. HOLE REPAIR

Holes can occur in an edge either as a result of noise, or some previous action taken by another correction algorithm (i.e., bubble repair). A hole is simply a discontinuity in an edge. Holes that occur as a result of noise are quite small, usually spanning only one or two pixels. On the other hand, holes that result from previous correction procedures may be many pixels wide.

We have implemented a *hole-repair algorithm*, which will draw a straight-line approximation of the edge to fill the hole. First, holes are located by using our edge tracking algorithm. The error condition of the edge tracker occurs when a hole is reached. Here, the edge tracker returns the list of edge points, up to the start location of the hole, and quits.

The hole-repair algorithm will then locate the end location of the hole by scanning beyond the hole start location in the direction of the edge until the opposite side of the hole is found. Next, the algorithm calculates the horizontal and vertical distance between the edge points bordering the hole. A straight line that connects the border points of the hole is then calculated and used to fill the hole defect.

3. DISTORTION ESTIMATION AND CORRECTION

With the quality checks and repair algorithms in place, the parameters of the distortion correction model can be estimated. The data is ensured to be free of nearly all defects and blemishes, which ensures the success and accuracy of the distortion correction. Linear least squares fitting methods are used to create the ideal phantom planes and to estimate the parameters of the distortion correction model, which is based on the sum of spherical harmonics. Once the model is known it can be applied to all studies performed on the particular scanner. The distortion correction method is discussed at length in [7].

4. RESULTS

Using the data processing methods described above, it was found that even image sets of relatively poor quality could be converted to useful image sets for use in subsequent data processing steps. For example, in the case of the study conducted with the leaking phantom, 50% of the edge images were corrupted by the oil leakage. Additionally, a large corner bubble was present in a good number of the edge images. After applying the techniques described above, *all* the edge images were successfully repaired. All traces of the large air bubble were removed, and the large hole in the bottom right quadrant was properly filled. The study was very suitable for further calculations.

Performing the distortion correction on the phantom's faces has been shown to have an accuracy of 0.12 to 0.25 mm, as small as about one-third of a pixel. Depending on the model of MRI scanner used, it may be possible for the scanner to apply image filtering techniques to *reduce* the effects of the distortion prior to processing by this software package. Since it is possible for a scan procedure to incorporate filtering techniques, testing of the distortion correction also involved processing images using the various filters available on the MRI scanner. The particular scanner utilized for software development employs three filtering options: a large FOV filter, which filters the entire image; an elliptical filter, which filters the centermost portions of an image; and no filtering. Regardless of the filter used by the MRI scanner, consistent correction results were produced in all test cases. Most surprising was the results obtained with scans performed using the large FOV filter. In these scans, the software was able to further improve upon the MRI scanner's distortion filter, even though the filtered images from the scanner already appeared to contain no distortion.

5. CONCLUSIONS AND FUTURE WORK

Recent testing has revealed two issues that affect the current accuracy of the correction. Although the current accuracy of the correction results is indeed submillimeter, there is still room for improvement. The theoretical accuracy of the correction has been identified to be one-third of a pixel, and efforts now concentrate on enhancing the accuracy of the system.

In-depth analysis of the original scan images revealed the presence of an additional distortion in particular scan sequences. Investigation of the acquired data showed evidence of a tilt in one of the phantom's faces (see Figure 5), which skewed the correction results in one dimension. Testing narrowed the possible sources to either asymmetrically-shaped phantom faces, or a discrepancy caused by the MRI scanner. After performing additional scans, the problem was identified to be the result of improperly calibrated magnets in the MRI scanner. Performing a magnet shimming procedure on the MRI scanner alleviated the problem, and removed the extra distortion. From this testing, it was deemed that the magnet shimming procedure is a necessary step to perform in order to maximize the accuracy of the correction.

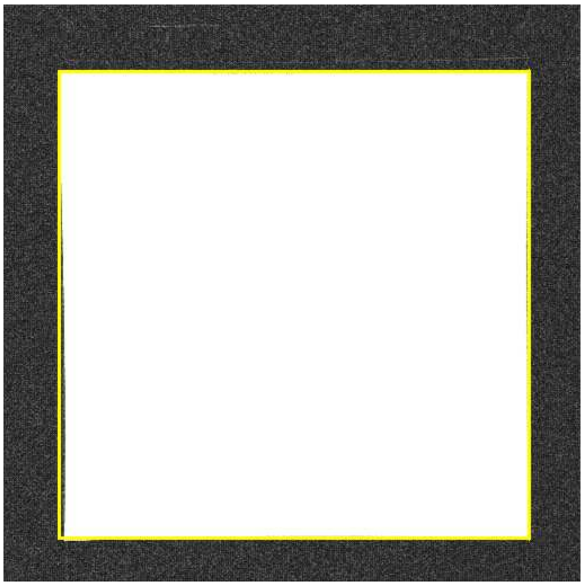


Fig. 5. Image with "tilted" face, apparent in bottom left corner. A yellow box is drawn around the image of the phantom to demonstrate tilting.

Testing also identified the need for a reliable method of centering the phantom within the MRI scanner, to ensure that the distortion in the images is as symmetric as possible. The software has identified the phantom to be offcentered in over half of the studies performed. Although the average displacement of the phantom was only a few millimeters from the center, the effects of the asymmetry in the distortion do indeed influence the outcome of the distortion correction. To solve this issue, the phantom was modified by etching straight lines into the faces. With these reference lines, it is now possible to align the phantom properly using the laser localizer sight, normally used to center the patient head coil. The sight has the ability to center the head coil with submillimeter precision; the same will hold true for the phantom as well. Utilizing this method will help ensure submillimeter precision of the distortion correction.

The current phase of testing has only analyzed the accuracy of the method using the original calibration phantom. Further verification will be conducted using other phantoms with known fiducial markers. The positions of the markers calculated by the software will be compared to the actual positions of the markers in the phantoms. The final phase of verification will test the system in a preclinical study by performing procedures on rats.

6. REFERENCES

[1] D. Kondziolka, "Functional radiosurgery," *Neurosurgery*, vol. 44, pp. 12–20, 1999.

[2] G. Bednarz, M.B. Downes, B.W. Corn, W.J. Curran, and H.W. Goldman, "Evaluation of the spatial accuracy of magnetic resonance imaging-based stereotactic target localization for gamma knife radiosurgery of functional disorders," *Neurosurgery*, vol. 45, pp. 1156–1161, 1999.

[3] S.J. Doran, L. Charles-Edwards, S.A. Reinsberg, and M.O. Leach MO, "A complete distortion correction for MR images: I. Gradient warp correction," *Phys. Med. Biol.*, vol. 50, pp. 1343–1361, 2005.

[4] D. Wang, W. Strugnell, G. Cowin, D.M. Doddrell, and R. Slaughter, "Geometric distortion in clinical MRI systems Part I: evaluation using a 3D phantom," *J. Magn. Reson. Imaging*, vol. 22, pp. 1211–1221, 2004.

[5] D. Wang, W. Strugnell, G. Cowin, D.M. Doddrell, and R. Slaughter, "Geometric distortion in clinical MRI systems Part II: correction using a 3D phantom," *J. Magn. Reson. Imaging*, vol. 22, pp. 1223–1232, 2004.

[6] S. Langlois, M. Desvignes, J.M. Constans, and M. Revenu, "MRI geometric distortion: a simple approach to correcting the effects of non-linear gradient fields," *J. Magn. Reson. Imaging*, vol. 9, pp. 821–831, 1999.

[7] T.S. Lee, K.E. Schubert, and R.W. Schulte, "Gradient Non-Linearity Correction of MR Images for Functional Radiosurgery," in *to appear in Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science*, 2006.