A New Method for Registration-Based Medical Image Interpolation

David H. Frakes*, Lakshmi P. Dasi, Kerem Pekkan, Hiroumi D. Kitajima, Kartik Sundareswaran, Ajit P. Yoganathan, and Mark J. T. Smith

***Abstract*—A new technique is presented for interpolating between grey-scale images in a medical data set. Registration between neighboring slices is achieved with a modified control grid interpolation algorithm that selectively accepts displacement field updates in a manner optimized for performance. A cubic interpolator is then applied to pixel intensities correlated by the displacement fields. Special considerations are made for efficiency, interpolation quality, and compression in the implementation of the algorithm. Experimental results show that the new method achieves good quality, while offering dramatic improvement in efficiency relative to the best competing method.**

***Index Terms*—Interpolation, reconstruction, registration.**

I. Introduction

THREE-DIMENSIONAL (3-D) medical imaging modalities often present acquired data as a set of slices. The thickness of each slice is usually significantly greater than the distance between voxel centers within an imaging plane, resulting in a data set composed of voxels that are anisotropic. In many applications that deal with 3-D data, it is desirable to have voxel dimensions that are isotropic or nearly so. These include multi-planar reconstruction (MPR), maximum intensity projection, and shaded surface rendering to name a few. Here, a novel interpolation technique for increasing the out-of-plane resolution of medical image data sets, analogous to decreasing slice thickness, is presented. The new methodology performs well in comparison to other state-of-the-art techniques based on quality and offers significant advantages in terms of both computational requirements and ease of implementation in commercial applications.

Interpolation methods for this type of problem can be generally classified as being scene-based or object-based [1]. Scene-based approaches use uniform registration, customarily interpolating between intensity values that are correlated based on their matrix locations within respective images. This category includes many well-known methods such as linear, spline, and truncated sinc function interpolation [2]. Several enhancements along these lines have also been proposed that make use of more complex convolution kernels [3], [4]. Scene-based techniques are efficient and easily implemented, but can produce significant artifacts when pixels that occupy the same matrix location in contiguous images belong to different anatomical structures.

Object-based interpolation techniques exploit information contained in the image slices to facilitate more accurate interpolation. This category can be further subdivided into interpolators that operate on extracted features or contours [5]–[9], and ones that operate directly on image intensity values [10]. In this paper, we are primarily concerned with the latter category, but also with special cases of the former category that allow whole-image interpolation, as that is the problem being addressed. Specific examples include the shape-based method proposed by Grevera and Udupa [11], the registration-based method proposed by Penney *et al.* [12], and a number of optical flow-based methods [13], [14].

An historical account of proposed solutions indicates that object-based methods have long been recognized as superior to scene-based methods in terms of quality. Nevertheless, most clinical software products that demand interslice interpolation still rely on scene-based methods. In particular, most picture archiving and communication systems (PACS) use either trilinear or tricubic interpolation to accomplish MPR. Why is this the case if clearly superior-quality methods are available? The answer lies in prohibitive computation time. The computational expense of interpolation is important in commercial applications as resources available on client machines vary and the time required to perform interpolation is often critical. Revisiting the previous MPR example, the time scale of approximately 7 min reported in [12] for interpolating a single computed tomography (CT) image would be unacceptable in today's PACS applications.

Since the computational requirements of object-based interpolation techniques can vary considerably, a second factor, compression, can affect the overall practicality of an approach. If interpolation can be carried out by a high-powered machine at a central location, then the implications of computational expense become less significant, but storage and transmission issues arise as interpolated data sets contain more images. For example, computing an isotropic CT data set can require the addition of more than 10 slices between each acquired one. Clearly the storage and transmission of all acquired and interpolated images, even with state-of-the-art lossless compression techniques, is not a viable option in an environment where storage

Manuscript received November 15, 2005; revised July 5, 2007. This work was supported by the U.S. National Institutes of Health, specifically the National Heart, Lung, and Blood Institute, under Grant R01HL67622. *Asterisk indicates corresponding author*.

*D. H. Frakes is with 4-D Imaging, Inc., Atlanta, GA 30332 USA and with the Wallace H. Coulter Department of Biomedical Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: frakules@mac.com).

L. P. Dasi, H. D. Kitajima, K. Sundareswaran, and A. P. Yoganathan are with the Wallace H. Coulter Department of Biomedical Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA.

K. Pekkan is with the Department of Biomedical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA.

M. J. T. Smith is with the Department of Electrical Engineering, Purdue University, West Lafayette, IN 47907 USA.

Digital Object Identifier 10.1109/TMI.2007.907324

0278-0062/$25.00 © 2007 IEEE

cost and transmission speed are factors. However, data interpolated with the proposed method can inherently be highly compressed, making this technique more practical in the central processing scenario.

A new object-based interpolation method based on an improved form of control grid interpolation (CGI) is presented here. The modified control grid interpolation (MCGI) is an optical-flow-based technique like standard CGI, but incorporates several improvements. It outperforms previous CGI implementations in terms of quality, and also performs favorably in comparison to other recently proposed object-based methods as the results to follow indicate. Furthermore, the low computational expense and high lossless compression associated with the new technique make it well suited for implementation in commercial medical imaging applications.

II. Methods

CGI is a hybrid motion estimation technique that incorporates features of both block-matching and optical flow. These two underlying approaches will not be presented here but have been covered extensively in the literature [13], [15], [16]. Like block-matching, CGI partitions images into subregions. The displacement vectors describing the movement of pixels from one image to the next are calculated within each subregion using the optical flow constraint equation

$$\frac{\partial I(x,y,t)}{\partial x}\frac{\partial x^*}{\partial t} + \frac{\partial I(x,y,t)}{\partial y}\frac{\partial y^*}{\partial t} + \frac{\partial I(x,y,t)}{\partial t} = 0 \qquad (1)$$

where $I(x,y,t)$ represents a grey-scale image and $x^*(t)$ and $y^*(t)$ indicate the true trajectory of each point within the image [15]. These displacement vectors are employed to register similar anatomical features in adjacent image slices. Interpolation is then performed between registered pixels. Specifics of the standard CGI algorithm are detailed in the following section to lay the foundation for the improved algorithm.

A. Conventional CGI

The conventional CGI implementation first divides an image into rectangular subregions and assumes that the pixel displacement functions within each region R can be written as

$$d_1(\mathbf{n}) = \sum_{i=1}^{p} \alpha_i \theta_i(\mathbf{n}) \qquad (2)$$

and

$$d_2(\mathbf{n}) = \sum_{i=1}^{p} \beta_i \phi_i(\mathbf{n}) \qquad (3)$$

where $\theta_i(\mathbf{n})$ and $\phi_i(\mathbf{n})$ are independent basis functions used to describe the displacement field, $\mathbf{n} = (n_1, n_2)$ and α_i and β_i are the displacement parameters to be estimated. Equivalently these displacement equations can be written in vector form as

$$d_1(\mathbf{n}) = \overline{\alpha}^T \overline{\theta}(\mathbf{n}) \qquad (4)$$

and

$$d_2(\mathbf{n}) = \overline{\beta}^T \overline{\phi}(\mathbf{n}) \qquad (5)$$

where $\overline{\alpha}, \overline{\beta}, \overline{\theta}$, and $\overline{\phi}$ denote vectors with elements $\alpha_i, \beta_i, \theta_i$, and ϕ_i, respectively. In this application, the basis functions within each region R can be written explicitly as

$$\theta_1(n_1,n_2) = \phi_1(n_1,n_2) = \left(\frac{n_1^2 - n_1}{n_1^2 - n_1^1}\right)\left(\frac{n_2^2 - n_2}{n_2^2 - n_2^1}\right) \qquad (6)$$

$$\theta_2(n_1,n_2) = \phi_2(n_1,n_2) = \left(\frac{n_1^2 - n_1}{n_1^2 - n_1^1}\right)\left(\frac{n_2 - n_2^1}{n_2^2 - n_2^1}\right) \qquad (7)$$

$$\theta_3(n_1,n_2) = \phi_3(n_1,n_2) = \left(\frac{n_1 - n_1^1}{n_1^2 - n_1^1}\right)\left(\frac{n_2^2 - n_2}{n_2^2 - n_2^1}\right) \qquad (8)$$

and

$$\theta_4(n_1,n_2) = \phi_4(n_1,n_2) = \left(\frac{n_1 - n_1^1}{n_1^2 - n_1^1}\right)\left(\frac{n_2 - n_2^1}{n_2^2 - n_2^1}\right) \qquad (9)$$

where (n_1^1, n_2^1) denotes the coordinates of the upper left corner of a rectangular region and (n_1^2, n_2^2) the coordinates of the lower right corner. It is noteworthy that at each corner of a region only one of these basis functions is equal to one, while the others have a value of zero. Function values throughout the region, which dictate the spatial variation of the displacement field, correspond to a bilinear model. Like the basis function vectors, the displacement parameter vectors $\overline{\alpha}$ and $\overline{\beta}$ each have four components. The motion model employed in this work is a connected model, meaning that the parameters associated with a given region's corner point are also valid for that point in the other three regions of which that point is a member. This relationship between neighboring regions contributes to displacement field smoothness as local displacements in separate regions are not totally independent and adjacent regions are guaranteed to be continuous. Global information is thus incorporated even into local displacement estimates and the entire resultant displacement field is piecewise smooth and globally continuous as one would expect given the typically smooth variation of anatomy.

In this formulation, the displacement vector at each pixel is represented as a linear sum of the previously outlined basis functions, which leads to a quadratic relationship between the approximate objective function and the model parameters. The model parameters from (2) and (3), in this case eight of them, must be estimated within each region of support. To do this efficiently, the optical flow equation at time $k + \delta k$ is represented with a first order Taylor series expansion. Making this substitution, the optical flow brightness constraint

$$I[n_1,n_2,k] = I\left(n_1 + d_1[n_1,n_2,k], n_2 + d_2[n_1,n_2,k], k + \delta k\right) \qquad (10)$$

becomes

$$I[\mathbf{n},k] \approx I[\mathbf{n},k+\delta k] + \frac{\partial I[\mathbf{n},k+\delta k]}{\partial n_1} d_1(\mathbf{n}) + \frac{\partial I[\mathbf{n},k+\delta k]}{\partial n_2} d_2(\mathbf{n}) \qquad (11)$$

where again $\mathbf{n} = (n_1, n_2)$. Thus, (11) provides a discrete-time approximation of the optical flow constraint equation in (1). Expressing this equation explicitly in terms of the model parameters from (4) and (5), we obtain

$$I[\mathbf{n},k] \approx I[\mathbf{n},k+\delta k] + \frac{\partial I[\mathbf{n},k+\delta k]}{\partial n_1}\overline{\alpha}^T\overline{\theta}(\mathbf{n}) + \frac{\partial I[\mathbf{n},k+\delta k]}{\partial n_2}\overline{\beta}^T\overline{\phi}(\mathbf{n}). \quad (12)$$

Equation (12) is assumed to hold for all pixels within R. In order to calculate the optimal displacement model parameters, the total squared error function

$$\widetilde{E}(\overline{\alpha},\overline{\beta}) = \sum_{\mathbf{n}\in R} \left(I[\mathbf{n},k] - I[\mathbf{n},k+\delta k] + \frac{\partial I[\mathbf{n},k+\delta k]}{\partial n_1}\overline{\alpha}^T\overline{\theta}(\mathbf{n}) + \frac{\partial I[\mathbf{n},k+\delta k]}{\partial n_2}\overline{\beta}^T\overline{\phi}(\mathbf{n})\right)^2 \quad (13)$$

is minimized. It is important to distinguish between the error associated with the displacement field in (13), and the error corresponding to interpolated values, as the two do not always correlate. In general, the latter is referred to in this text as interpolation error and corresponds to the mean squared difference measure that is defined later as (14). Repeating this process for each region R within an image yields a displacement field that registers each pixel to some location in the neighboring image. The process can be repeated for all regions, using the result of the first pass as a starting point, to decrease the error in (13). The gain realized for further and further passes in terms of error reduction is usually small in comparison to the increased computation time, however.

Using one complete displacement field and the image it originates from, an intermediate image can be reconstructed. Each discrete location on the intermediate image grid is bounded by the intersections of four independent displacement vectors with that intermediate plane. These vectors can be identified easily by rounding. The pixel intensity at a discrete location in the intermediate image is determined based on a linear interpolation involving the intensities at the origins and destinations of the four bounding vectors. There are rare cases when rounding does not reveal the immediate four bounding vectors, or when the intersection of a displacement vector with the intermediate image plane is coincident with a discrete location, but these instances are easy to identify and handle on a case-by-case basis. For a given image pair there are two displacement fields: one reflecting pixel displacements from the first image of the pair to the second, and the other reflecting displacements from the second image of the pair to the first. The reconstructed images corresponding to the respective displacement fields are combined in a spatially weighted sum to produce the final interpolated image.

It is noteworthy that unlike some other techniques, the methodology presented here does not create, as an intermediary, a single registration that serves as a bidirectional mapping between two images [17]. In other words, the displacement field mapping image A to image B will not be the reverse of the field mapping image B to image A. Consequently, the registration component of the proposed algorithm is not symmetric. However, the interpolation process as a whole is directionally unbiased. This is because each interslice interpolated image is formed as a weighted sum of forward and backward approximations of that image.

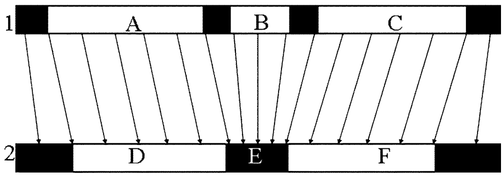

Fig. 1. Simple example illustrating the capabilities of connected motion models in the context of data interpolation. Here, the high intensity region B disappears during the transition from plane 1 to plane 2. However, since the displacement field is calculated to minimize (13), the vectors emanating from B converge and the interpolation errors manifested by the disappearance of B are less widespread than for a scene-based approach corresponding to zero value displacement vectors.

The general CGI formulation, the algorithm proposed here, and the algorithm in [12] that is cited frequently in this work for comparison purposes, are all registration-based interpolation algorithms. Prior to the actual interpolation phase, a displacement field is determined that links an image to its adjacent neighbor. Registration-based algorithms rely on two assumptions: first that adjacent slices contain similar anatomic features, and second that the registration algorithm in use is capable of determining the appropriate geometric transform to link them. When either of these two assumptions is violated, the advantages of registration-based techniques diminish.

Violation of the first assumption does occur repeatedly when an anatomical structure disappears from one slice to the next, or when branching causes a structure to appear very differently when sampled by adjacent slices. When these cases are present a good solution to the interpolation problem may not exist. That is, no interpolation algorithm can provide an accurate solution when the information required to solve the ambiguity is unavailable. However, when an anatomical feature does disappear or branch, the displacement vectors from neighboring control regions can be used effectively to influence the displacement field in the problematic region so as to minimize the interpolation error that is realized. This ability is an advantage of connected motion models, and is illustrated by the simple example in Fig. 1. Here, the high-intensity region labeled B disappears between planes 1 and 2. However, the motion model matches region A to region D, and region C to region F, in order to minimize error. The result is that the interpolation vectors emanating from region B converge. In this way, the connected model allows other displacements within the plane that correspond to strong matches to influence regions where the correct solution is more ambiguous given the available information. Similar effects characterize the performance of the algorithm when branching is present. Since the displacement field typically varies smoothly for real anatomies, model connection enables more accurate interpolation even in areas where CGI's optical flow foundation does not contribute directly to an accurate solution. Violation of the second assumption, given that the first is upheld, is very infrequent as optical flow is capable of characterizing highly complex motion. The lower bound on control point resolution used here (i.e., each separated from the next by two pixels) ensures that this capability is exploited effectively.

B. Modifications

The proposed algorithm differs from traditional CGI implementations in three primary ways. First, in the modified method the CGI framework is applied recursively to decimated versions of the images being analyzed, and displacement field solutions are retained only when specific criteria are met. Second, displacement vector precision for the modified CGI implementation is constrained to ensure that the resultant displacement field solution can be stored efficiently. This constraint leads to high lossless interpolated slice compression and to decreased computation. Lastly, multiple image pairs are selectively incorporated into the interpolation phase to allow higher order interpolation to be performed, as opposed to linear interpolation, which is common to most registration-based techniques. Each of these distinguishing factors will be covered in more detail in the following paragraphs.

1) Adaptive Optimization Framework: The standard CGI formulation has been used to address many problems in the past [18]–[20]. Typically a large block size is used initially and progressively smaller block sizes are then used in a static or adaptive framework to obtain a higher resolution displacement field. Here, we take a slightly different approach, applying a static block size to progressively less decimated images to achieve a similar effect. Decimation is achieved in this application with linear block averaging. In practice neither method produces consistently better quality, but the approach taken here demands less computation as the gradient calculations required for each iteration are carried out on the smaller decimated images until the last step. This method is similar to some multi-grid optical flow techniques that have been proposed previously, except that the smoothness constraint is imposed by the motion model rather than with an additional term [15], [21]. Additionally, the block size is never taken down to the single pixel level as in complete multigrid optical flow approaches, which involve greater computation.

The primary goal of any interslice interpolation algorithm is to reconstruct the information between acquired slices as accurately as possible. The more immediate goal of this MCGI implementation, and any other optimization-based interpolation technique that makes use of registration, is to optimize a cost function in order to facilitate accurate interpolation. However, the cost function [(13) in this case], while strongly related to interpolation error, is not a true measure of interpolation accuracy. Successive iterations of conventional CGI that decrease the error in (13) do not always increase interpolation accuracy as measured against ground truth. Evidence of this was observed repeatedly in experimental trials (with ground truth) involving more than 200 data sets, from which empirical trends were noted. These trends and statistics in turn were used to further enhance the performance of the MCGI. Using the displacement field solution for the highly decimated images as a starting point for the less decimated images often leads to increased interpolation error. Empirically, we have found that if the error in (13) is not decreased by 50% or more for the first iteration, better interpolation results are usually obtained by skipping directly to the last iteration and using a zero displacement field as the starting point. In such cases, large-scale displacements are not significantly present and starting with a clean slate to characterize the smaller-scale displacements works best. Likewise when (13) is decreased by less than 10% for any iteration, it usually does not contribute to more accurate interpolation, so the displacement field update is rejected. Again, the 50% iteration threshold parameter and the 10% termination threshold parameter reflect experimental statistics. Each parameter was changed in 5% increments in the analysis. The iteration parameter was varied from 10% to 70% and the termination parameter from 0% to 30%. The average interpolation error values for all of the data sets were then used to select the best parameter values. Clear trends in the variation of interpolation error were present.

To substantiate the claim that these modifications to conventional CGI do in fact improve quality, a validation was performed using the data sets presented later in Section III. As opposed to following the conventional CGI framework, whereby all displacement field updates are accepted until (13) no longer decreases, the modified framework reduced true mean squared interpolation error by an average of 8.8%. This study was conducted independent of the other algorithmic modifications in order to demonstrate the effects of this modification alone.

2) Displacement Field Precision Constraint: The selective acceptance of displacement field updates, in addition to facilitating higher quality, allows the displacement field to be calculated more efficiently than with the general framework since iterations are skipped when the update criteria are not met. The second modification in the new algorithm, which also contributes to decreased computation, is the precision limitation that is imposed on the displacement field. Since displacement vector values are constrained to integers of absolute value less than 32 pixels in this implementation, the error minimization that is performed (a conjugate gradient algorithm) requires fewer iterations to complete. Furthermore, since each vector can be represented with 6 bits, efficient representation of the collective displacement field is ensured. This translates to high lossless compression for the interpolated slices as they can be reconstructed from the original slices and displacement fields alone. The benefits of using integer precision include a simple compact displacement field representation, more efficient compression, and more efficient processing. However, it should be noted that with or without the integer constraint, the MCGI computational requirements are relatively low and manageable from a practical perspective. A slightly lower interpolation error can be achieved with subpixel-accurate displacement vectors, but the displacement field cannot be represented as efficiently.

To demonstrate the effects of the precision constraint, another exercise was performed using the data sets presented later in Section III. Interpolation was performed both with the precision constraint imposed and without, i.e., with floating point precision. On average, imposing the constraint resulted in an interpolation error increase of 4.3% and a computation time decrease of 110.7%. As in the validation performed for the framework modification, this exercise was conducted independent of other modifications to show only the effects of the precision constraint. It is interesting to note that increased displacement vector accuracy does help decrease interpolation error. This is a well-established expectation in video compression (MPEG) applications. However, with respect to MCGI the gains from

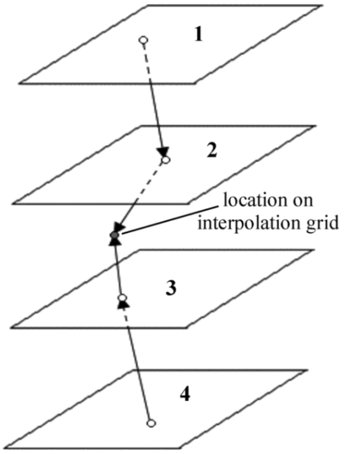

Fig. 2. Illustration of CGI modification allowing slices in addition to the neighboring pair to be incorporated into the interpolation process. Selective use of this strategy allows the basic interpolator to be upgraded from a linear to a cubic one. Previous registration-based algorithm have employed only linear interpolars.

higher precision are much less pronounced. To a large extent, this is because of the inherent inexactness associated with using the displacement error in (13), as mentioned earlier. That is, the higher precision components of the displacement field vectors that are lost to integer precision may be contributed by, in many cases, the later iterations of the algorithm that were found to increase interpolation error. As a result, precision related gains in quality are modest for MCGI. Therefore, it can be attractive to trade displacement vector precision for gains in computational efficiency. For this reason, integer precision displacement vectors were used in all subsequent implementations reported in this paper.

3) Use of Additional Displacement Fields: The last improvement that has been made to the traditional CGI formulation is the incorporation of adjacent displacement fields into the interpolation phase. Adjacent displacement fields can be linked to use data from slices outside the immediate pair of images when performing interpolation. For example, this process would involve pixel intensities from images 1–4 when interpolating between images 2 and 3, as indicated in Fig. 2. Performance gains are not realized in all cases, so this process is also used selectively. If a prospective additional intensity to be included differs from the neighboring value it is linked to by more than 20%, the standard linear interpolation is performed. In these cases, the best fit for a displacement field is still likely linking somewhat unrelated anatomical structures, and as such it is beneficial to use only the pixels in close proximity for interpolating. As with the other experimental parameters discussed in this section, the 20% threshold was determined through a search based on the aforementioned database of more than 200 data sets. Again the average interpolation errors for all of the data sets were used as selection criteria. The threshold was varied in 5% increments from 0% to 50% in order to identify the optimal value. When acceptable pixel intensity variations are present, the standard linear interpolation involving samples from planes 2 and 3 is replaced by a piecewise cubic hermite interpolation [22] using those samples and ones from planes 1–4, all of which have been linked by the displacement fields.

At the general level, the proposed reconstruction algorithm is markedly different from virtually all the methods reported previously by other authors, in that a dense and accurate displacement field is derived within a quadtree structure and displacement vectors are determined efficiently by optimizing a constrained optical flow cost function. Building on this novel framework, a number of specific implementation innovations are introduced. First, the difference between minimizing the displacement error and minimizing the interpolation error is recognized and exploited by selectively and adaptively retaining displacement vector information. Displacement vector calculations are performed within a multiresolution framework, which improves implementation speed. Along with this an adaptive integration of multislice prediction is employed to broaden the region of support over which estimation is performed. The new algorithm was designed as a complete system and engineered with computational efficiency (including integer vector representation) that is attractive for commercial implementation. The impact of the new algorithm on interslice reconstruction is made clear in the next sections. Experimental results show dramatic improvements over recently proposed registration-based techniques [12], [23], [24].

III. Experimental Validation

Evaluation of the MCGI algorithm was carried out in conformity with the previous evaluations conducted by Grevera and Udupa [11] and more specifically Penney *et al.* [12]. That is, the new method (mcgi) was compared against linear interpolation (lin) as a baseline, and then to a state-of-the-art algorithm from recent literature, in this case registration-based interpolation (reg) [12]. The registration-based algorithm selected for comparison here has been shown to outperform numerous other techniques in recent literature within the same evaluation framework. To help make straightforward comparisons, we have adopted the evaluation methodology used in [12].

A. Data

Interpolation was performed on four classes of images that were from two sites (head and chest) and two modalities [computed tomography (CT) and magnetic resonance (MR)] common to the evaluations performed in both [11] and [12]. Each of the four data classes was comprised of four individual data sets. Example images are provided in Fig. 3 and relevant imaging parameters are listed in Table I. To facilitate ongoing research along these lines, these data have been made publicly available so that future comparisons can be made conveniently. Details on how to acquire the data are given in the Acknowledgements section.

B. Methods

Each slice within the respective data sets, with the exception of the first and last, was removed one at a time. The MCGI algorithm was then applied to produce an interpolated version of each removed slice. Interpolated slices were compared directly to the removed slices using two error measures.

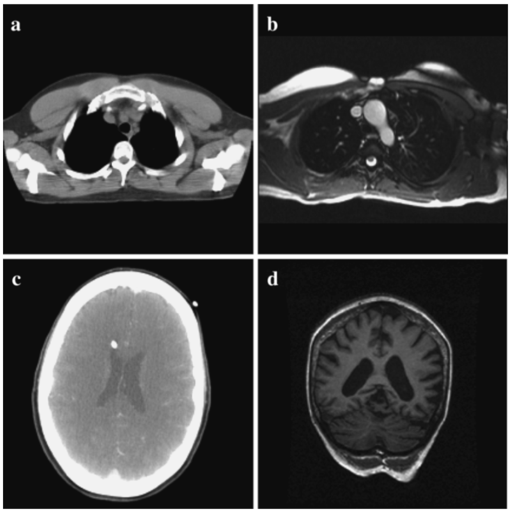

Fig. 3. Examples of medical images included in the interpolation evaluation. Images are from the (a) CT Chest, (b) MR Chest, (c) CT Head, and (d) MR Head data sets.

TABLE I
IMAGING PARAMETERS ASSOCIATED WITH EACH OF THE FOUR DATA CLASSES USED FOR VALIDATION

Data Class	Average # of Slices	Slice Thickness	Matrix Size (Pixels)	Pixel Dimensions (Square)
CT Head	41	3 - 3.5 mm	512 x 512	.40 - .57 mm
CT Chest	61	2.5 - 5 mm	512 x 512	.72 - .78 mm
MR Head	43	2.1 - 3.2 mm	256 x 256 - 512 x 512	.35 - .98 mm
MR Chest	30	3 - 4 mm	128 x 256 - 256 x 256	.78 - 1.17 mm

TABLE II
STATISTICAL RELEVANCE MEASURES COMPARING THREE DIFFERENT INTERPOLATION ALGORITHM BASED ON FOUR CLASSES OF DATA. DASH INDICATES THAT THE MEASURE WAS NOT STATISTICALLY SIGNIFICANT (T-TEST $p <= 0.05$)

Interpolation Methods	Data Class	MSD	NSD
reg/lin	CT Head	37.0	42.6
reg/lin	CT Chest	27.2	37.2
reg/lin	MR Head	23.4	18.9
reg/lin	MR Chest	34.7	37.9
mcgi/lin	CT Head	43.8	40.3
mcgi/lin	CT Chest	31.2	35.4
mcgi/lin	MR Head	28.1	15.9
mcgi/lin	MR Chest	34.7	27.9
mcgi/reg	CT Head	10.7	-
mcgi/reg	CT Chest	5.5	-
mcgi/reg	MR Head	6.1	-
mcgi/reg	MR Chest	-	-13.9

1) Mean squared difference (MSD) is expressed as

$$\mathrm{MSD} = \frac{1}{N_{\mathrm{sl}}} \sum_{i=1}^{N_{\mathrm{sl}}} \frac{1}{N_R} \sum_{\mathbf{u} \in R} \left(I_i(\mathbf{u}_{\mathrm{int}}) - I_i(\mathbf{u}_{\mathrm{rem}})\right)^2 \tag{14}$$

where $I_i(\mathbf{u}_{\mathrm{int}})$ and $I_i(\mathbf{u}_{\mathrm{rem}})$ represent the intensity in slice I at pixel position $\mathbf{u}$ in the interpolated and removed slices, respectively. N_{sl} represents the number of interpolated slices and N_R the number of pixels in each slice over which the error measure is summed. The region R is comprised of all pixels within a slice with the exception of those where data values are not present. The same region R was used to evaluate the different interpolation techniques in each respective case.

2) Number of sites of disagreement (NSD) is expressed as

$$\mathrm{NSD} = \sum_{i=1}^{N_{\mathrm{sl}}} \sum_{\mathbf{u} \in R} \tau\left(\|I_i(\mathbf{u})_{\mathrm{int}} - I_i(\mathbf{u})_{\mathrm{rem}}\|\right) \tag{15}$$

where

$$\tau(x) = \left\{ \begin{array}{ll} 1, & \text{if} \quad x > 0.05 \\ 0, & \text{otherwise} \end{array} \right\}. \tag{16}$$

This measure indicates the number of pixels where the difference between $I_i(\mathbf{u}_{\mathrm{int}})$ and $I_i(\mathbf{u}_{\mathrm{rem}})$ is greater than a specified threshold. In keeping with the conventions of the previous evaluations in [11] and [12], a threshold of 5% of the maximum intensity in the removed slice was used. The statistical relevance measure used in [11] and [12] was also employed here to quantify the difference between two interpolation techniques for a given error measure. For example, the statistical relevance r between MSD for linear interpolation, $\mathrm{MSD}_{\mathrm{lin}}$, and that for MCGI, $\mathrm{MSD}_{\mathrm{mcgi}}$, is

$$r_{\mathrm{cgi/lin}} = \left\{ \begin{array}{ll} +100 * \left[1 - \frac{\mathrm{MSD}_{\mathrm{mcgi}}}{\mathrm{MSD}_{\mathrm{lin}}}\right], & \text{if} \quad \mathrm{MSD}_{\mathrm{lin}} > \mathrm{MSD}_{\mathrm{mcgi}} \\ -100 * \left[1 - \frac{\mathrm{MSD}_{\mathrm{lin}}}{\mathrm{MSD}_{\mathrm{mcgi}}}\right], & \text{if} \quad \mathrm{MSD}_{\mathrm{mcgi}} > \mathrm{MSD}_{\mathrm{lin}} \end{array} \right\}. \tag{17}$$

IV. RESULTS

Table II shows mean statistical relevance values for the two error measures (MSD and NSD) that compare the three different interpolators: linear, registration-based, and MCGI. It is noteworthy that a third measure, largest disagreement (LD), was used in [11] and [12], but as in those papers the majority of the LD values here were statistically insignificant. Accordingly, the authors chose to focus on the two more meaningful metrics. The values in Table II represent averages computed from all of the images comprising each class of image data (CT Head, CT Chest, MR Head, and MR Chest). A dash in the table indicates that the difference between the interpolation methods being compared was not statistically significant (T-Test $p <= 0.05$).

The results in Table II indicate that both the registration-based technique and MCGI perform better than linear interpolation based on both error measures for all data classes. Using linear interpolation as a baseline, the proposed MCGI algorithm performs better than or equivalently to the registration-based technique for all data classes based on MSD. Using the same baseline, registration-based interpolation performs better for all data classes based on NSD. By direct comparison MCGI outperforms registration-based interpolation to a statistically significant degree on three occasions, while the latter technique is better only in one case. Half of the error measure outcomes for the direct comparison were statistically insignificant. A visual

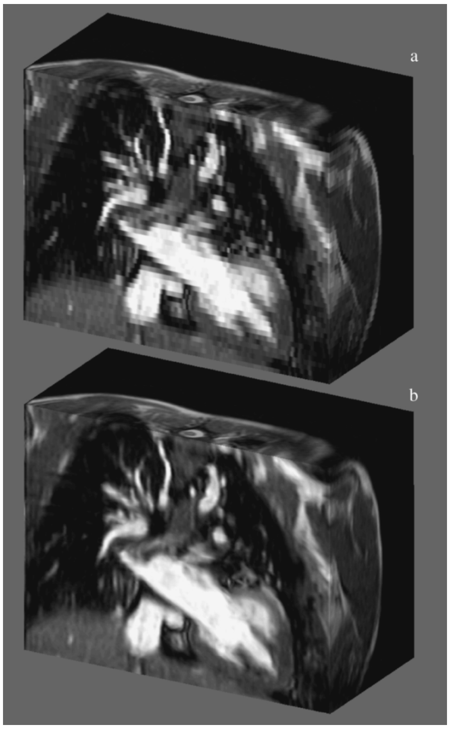

Fig. 4. Coronal MPR views of (a) original MRI data and (b) MCGI interpolated data. Reconstruction process for interpolating these data, following displacement field calculations, required less than 1 min.

TABLE III
CPU Times Required to Interpolate a Single 320 × 320 Pixel CT Image With a Slice Thickness of 5 mm and Pixel Dimensions of 0.75 ×0.75 mm. All Times Were Evaluated on PC-Based Systems With CPU Speeds Ranging From 1.5–2.0 Ghz

Algorithm	lin	reg	mcgi
CPU Time	0.13 secs	405 secs	4.82 secs

example of a data set reconstructed with MCGI is provided later in Fig. 4.

Table III lists the CPU time requirements for interpolating a single image using each of the three interpolation strategies that were evaluated. While MCGI performs slightly better or comparably to the algorithm in [12] based on error measures, its computational efficiency is dramatically better. The processing time required for MCGI to interpolate a single image is nearly two orders of magnitude less. Time measures reported here were quantified using a code profiler (MATLAB, MathWorks, Natick, MA).

V. Discussion and Conclusion

The results presented in the previous section show that in the majority of statistically significant cases MCGI performed better than the best reported registration-based method [12] based on interpolation error. The algorithm also outperformed previous CGI implementations including [14] based on the same metric. The dramatic advantage of the proposed methodology is speed of execution. Processing times nearly two orders of magnitude less were observed for the new algorithm as compared to the benchmark [12], making MCGI attractive for interslice reconstruction in many scenarios. Furthermore, unlike the registration-based method [12], MCGI reconstruction is directionally unbiased. It does not matter whether a stack of images is interpolated from top to bottom or vice versa; interpolations will be identical regardless.

The experimental comparisons can be appreciated at a more insightful level by considering the similarities and differences between MCGI and the method in [12]. With respect to the registration phases, the two have several similarities. Both use grid structures, have a comparable lower bound on grid resolution (10 mm for [12] versus approximately 7 mm for the proposed), and make use of intensity derivatives at similar intervals (0.625 mm at finest stage for [12] versus approximately 0.75 mm for the proposed). Both also employ forms of a coarse-to-fine recursive strategy to converge upon more accurate solutions. Given these common features, one would expect to see the fairly strong agreement in terms of quality performance that was observed. One fundamental difference between the two with respect to registration is the optimization criterion employed; a normalized mutual information measure is used in [12] while the proposed technique relies on a variation of the optical flow equation in conjunction with dynamic parameters aimed at optimizing interpolation quality. Apart from the registration phase, distinctive features of the proposed algorithm are directional symmetry (averaging reconstructions from both forward and backward predictions increases SNR and mitigates the effects of outliers), the ability to incorporate higher order interpolation, and the use of multiple reduced-precision displacement vectors in the postregistration intensity interpolation phase.

The most striking difference with respect to MCGI is computational efficiency. Three specific areas in which the new algorithm gains an advantage are in the iterative registration framework, the optimization routine within the registration phase, and the postregistration intensity interpolation phase. The specifics of these components of the proposed algorithm, and the advantages they offer, are covered in detail in Section II. To test the hypothesis that the aforementioned features were primarily responsible for the efficiency gains, modifications were made to the proposed algorithm to implement the mechanics presented in [12]. Following these modifications, the average execution time ballooned to 451 s, less than 10% removed from the reported execution time in [12], which validates the significance of these features with respect to computational efficiency.

Exploiting the inherent lossless compressibility of interpolated images is another potential advantage of MCGI, relevant for particular applications involving storage or transmission. For a single uncompressed 512 × 512 16-bit interpolated image that demands 524 288 B of space, the displacement fields used to derive it require approximately 36 552 B, assuming that they too are uncompressed. If multiple slices are to be added between each original one, no additional displacement fields are required as compared to adding a single slice. When this is the case, the effective compression ratio improves dramatically, scaling linearly with the additional number of images to be added.

Consider a real-world example involving a data set consisting of 200 512 × 512 16-bit CT images that can be losslessly compressed at a ratio of approximately 3:1 with wavelet methods.

Adding the displacement fields required to interpolate this data set to isotropy increases the total amount of data by only 8% after simple arithmetic coding. In this scenario, the only computation required to build the isotropic data set is the reconstruction of the interpolated images based on the originals and the corresponding displacement fields. This process is far less computationally expensive than regenerating the displacement fields as part of the decompression process. An average of 0.297 s is required for a 256 × 256 slice, putting the collective interpolation time into the range of minutes even for very large data sets. For example, the raw data set shown in Fig. 4(a), consisting of 50 256 × 256 MR images, was reconstructed to isotropy as in Fig. 4(b) in approximately 1 min. This magnitude of delay is still problematic for some applications, but is a drastic improvement and a step in the right direction. In comparison, the method in [12] reported processing times that would equate to nearly 6 h for the interpolation of 50 320 × 320 pixel CT images. Since no efficient representation of the interpolated images accompanies the other techniques evaluated here, the only alternative in this context would be to store or transmit all of the interpolated slices with the originals, increasing the total data set size by more than ten-fold in some cases.

Numerous clinical visualization tools such as MPR, maximum intensity projection, and shaded surface rendering represent 3-D information and perform best when high-quality isotropic data sets are available. These applications have become common practice in recent years and are often integrated into standard imaging protocols. The proposed algorithm can add value to these tools and to others that comprise the reconstruction to isotropy of multislice data sets. In the past, these applications have been unable to take advantage of more advanced interpolation methods due to prohibitive computation times. The algorithm presented here makes the use of advanced interpolation in the clinical environment more feasible.

In the proposed MCGI implementation, the modified optimization framework and postregistration interpolation phase contribute to higher quality and improved efficiency. Constraining displacement precision decreases computation with minimal quality loss and ensures efficient representation of the displacement field. Lastly, the use of neighboring displacement fields to enable higher order interpolation leads to better quality. These improvements contribute to an algorithm that represents a substantial improvement over previous registration-based methods and other state-of-the-art techniques aimed at medical image interpolation.

Acknowledgment

The Children's Hospital of Philadelphia, Children's Healthcare of Atlanta, Emory University, and the University of North Carolina at Chapel Hill contributed significantly to this research. The authors would like to thank to Dr. G. Penney of University College London. Without his assistance, the algorithmic comparisons presented here would not have been possible. To encourage future research, the image data that were used in this manuscript are available at http://homepage.mac.com/frakules/medicalimagedata.htm.

References

[1] G. J. Grevera and J. K. Udupa, "An objective comparison of 3-D image interpolation methods," *IEEE Trans. Med. Imag.*, vol. 17, no. 4, pp. 642–652, Aug. 1998.

[2] R. A. Zoroofi, Y. Sato, T. Sasama, T. Nishii, N. Sugano, K. Yonenobu, H. Yoshikawa, T. Ochi, and S. Tamura, "Automated segmentation of acetabulum and femoral head from 3-D CT images," *IEEE Trans. Inf. Technol. Biomed.*, vol. 7, no. 4, pp. 329–343, Dec. 2003.

[3] E. H. W. Meijering, W. J. Niessen, and M. A. Viergever, "Quantitative evaluation of convolution-based methods for medical image interpolation," *Med. Image Anal.*, vol. 5, no. 2, pp. 111–126, 2001.

[4] P. Thevanaz, T. Blu, and M. Unser, "Interpolation revisited," *IEEE Trans. Med. Imag.*, vol. 19, no. 7, pp. 739–758, Jul. 2000.

[5] T. Y. Lee and C. H. Lin, "Feature-guided shape-based image interpolation," *IEEE Trans. Med. Imag.*, vol. 21, no. 12, pp. 1479–1489, Dec. 2002.

[6] T. Y. Lee and W. H. Wang, "Morphology-based three-dimensional interpolation," *IEEE Trans. Med. Imag.*, vol. 19, no. 7, pp. 711–721, Jul. 2000.

[7] G. Herman, J. Zheng, and C. A. Bucholtz, "Shape-based interpolation," *IEEE Comput. Graph. Appl.*, vol. 12, no. 3, pp. 69–79, May 1992.

[8] S. P. Raya and J. K. Udupa, "Shape-based interpolation of multidimensional objects," *IEEE Trans. Med. Imag.*, vol. 9, no. 1, pp. 32–42, Mar. 1990.

[9] W. E. Higgins, C. Morice, and E. L. Ritman, "Shape-based interpolation of tree-like structures in three-dimensional images," *IEEE Trans. Med. Imag.*, vol. 12, no. 3, pp. 439–450, Sep. 1993.

[10] A. Goshatsby, D. A. Turner, and V. Ackerman, "Matching of tomographic slices for interpolation," *IEEE Trans. Med. Imag.*, vol. 11, no. 4, pp. 507–516, Dec. 1992.

[11] G. J. Grevera and J. K. Udupa, "Shape-based interpolation of multidimensional grey-level images," *IEEE Trans. Med. Imag.*, vol. 15, no. 6, pp. 881–892, Dec. 1996.

[12] G. P. Penney, J. A. Schnabel, D. Rueckert, M. A. Viergever, and W. J. Niessen, "Registration-based interpolation," *IEEE Trans. Med. Imag.*, vol. 23, no. 7, pp. 922–926, Jul. 2004.

[13] W. L. Williams and W. A. Barrett, "Optical flow interpolation of serial slice images," *Proc. SPIE Med. Imag.*, vol. 1898, pp. 93–104, 1993.

[14] D. H. Frakes, C. Conrad, T. Healy, J. W. Monaco, M. J. T. Smith, M. Fogel, S. Sharma, and A. P. Yoganathan, "Application of an adaptive control grid interpolation technique to morphological vascular reconstruction," *IEEE Trans. Biomed. Eng.*, vol. 50, no. 2, pp. 197–206, Feb. 2003.

[15] J. W. Monaco, "Motion models for video applications," Ph.D. dissertation, Georgia Inst. Technol., Atlanta, Nov. 1997.

[16] B. Horn, *Robot Vision*. Cambridge: MIT Press, 1968, pp. 280–292.

[17] P. Cachier and D. Rey, "Symmetrization of the non-rigid registration problem using inversion-invariant energies: Application to multiple sclerosis," in *Proc. MICCAI 2000*, Oct. 2000, pp. 472–481.

[18] A. M. Tekalp, *Digital Video Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[19] M. J. Black and P. Anandan, The robust estimation of multiple motions: Affine and piecewise-smooth flow fields Xerox Systems Practices Lab., Palo Alto, CA, 1993.

[20] R. Szeliski and H. Y. Shum, "Motion estimation with quadtree splines," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 12, pp. 1199–1210, Dec. 1996.

[21] W. Zhao and H. Sawhney, "Is super-resolution with optical flow feasible?," in *Proc. Eur. Conf. Comput. Vis.*, 2002, vol. 1, pp. 599–613.

[22] O. Niemitalo, Polynomial interpolators for high-quality resampling of audio 2001 [Online]. Available: http://www.student.oulu.fi/~oniemita/DSP/

[23] N. Pagoulatos, W. S. Edwards, D. R. Haynor, and Y. Kim, "Interactive 3-D registration of ultrasound and magnetic resonance images based on a magnetic position sensor," *IEEE Trans. Inf. Technol. Biomed.*, vol. 3, no. 4, pp. 278–288, Dec. 1999.

[24] C. R. Castro-Pareja, J. M. Jagadeesh, and R. Shekhar, "FAIR: A hardware architecture for real-time 3-D image registration," *IEEE Trans. Inf. Technol. Biomed.*, vol. 7, no. 4, pp. 426–434, Dec. 2003.