Flow visualization for qualitative assessment of brain shift

Lyubomir Zagorchev[a], Ardeshir Goshtasby[a], Martin Satter[b]

[a]CSE Department, Wright State University, Dayton, OH 45435
[b]PET Department, Kettering Medical Center, Kettering, OH 45429

ABSTRACT

A method for detection, quantification, and visualization of brain shift in serial MR and CT images is presented. The method consists of three steps. It first establishes correspondence between a number of point landmarks in the images. It then uses the correspondences to determine a transformation function that warps one image to the geometry of the other. It finally uses the obtained transformation to create a vector flow that represents the local motion or deformation of one image with respect to the other. The method does not require the solution of a system of equations and, therefore, is especially effective when a large number of correspondences is needed to represent complex brain deformations.

Keywords: flow visualization, nonrigid registration, brain shift, image warping

1. INTRODUCTION

Frequently, growth or removal of a tumor in the brain results in deformation or local movement of the brain, known as brain shift. Visualization of brain shift during surgery allows the surgeon to assess the progression of surgery and take corrective actions, if necessary. Brain shift can result from a number of surgical operations, such as tumor resection, cerebrospinal fluid drainage, hyperventilation, edema, and hemorrhage. As the operation progresses, the amount and likelihood of brain shift increase. Brain shift, if not compensated for, can result in surgical inaccuracies and consequently damage to the patient.

Studies to quantify brain shift have been attempted before. Bucholz et al.[1] and Koivunkangas et al.[2] registered intraoprative ultrasound images with preoperative MR images to quantify brain shift. Dickhaus et al.[3] quantified brain shift by tracking point landmarks in pre- and intraoperative MR images. Roberts et al.[4] quantified brain shift by tracking features in color images taken at the start and during surgery by a camera positioned above the opening of the brain. The work presented here describes a visualization tool that can aid a surgeon to qualitatively assess brain shift during surgery.

Visualization of brain shift is achieved through nonrigid image registration. Nonrigid registrations usually use nonlinear transformation functions such as polynomials,[5] splines,[6–8] and multiquadrics.[9–11] These are interpolative functions that require the solution of systems of equations and map corresponding landmarks in the images to each other exactly. Such functions work well when spacing between the landmarks is rather uniform and a small number of them (up to a few hundred) is available. When spacing between the landmarks varies considerably across the image domain and/or thousands of landmarks are available, interpolating transformation function often produce large errors away from the landmarks. A class of transformation functions[10, 11] that maps corresponding landmarks to each other approximately does not require the solution of a system of equations and, thus, is more stable under variations in spacing between the landmarks. Also, because it does not require the solution of a system of equations, it work well with small as well as large landmark correspondences. In addition, the approximation process smoothes noise often observed among the correspondences.

In this paper, a visualization method for qualitative assessment of brain shift using an approximating transformation function is presented. The method consists of three steps. In the first step, a set of corresponding landmarks in the images is found. In the second step, the correspondences are used to determine a transformation function that maps the rest of the image points to each other. The method does not use any assumptions about

Further author information: Lyubomir Zagorchev is currently with Philips Research North America, Briarcliff Manor, NY 10510, E-mail: Lyubomir.Zagorchev@philips.com, Telephone: (914) 945-6138. Send correspondence to: Ardeshir Goshtasby, E-mail: agoshtas@wright.edu, Telephone: (937)775-5170

Medical Imaging 2006: Visualization, Image-Guided Procedures, and Display,
edited by Kevin R. Cleary, Robert L. Galloway, Jr., Proc. of SPIE Vol. 6141, 614127, (2006)
1605-7422/06/$15 · doi: 10.1117/12.649900

the arrangement of the landmarks. The method can work with large landmark correspondences whose densities vary across the image domain. The computed transformation function is then used to create a vector flow that reflects the amount and direction of local motion of one image points with respect to the other. In this paper, we will call one of the images the *reference* and the second image the *target*. Deformation of the target image with respect to the reference image is determined and visualized.

2. METHOD

A method for visualizing brain shift quantified through a nonrigid registration is described. Nonrigid registration has been achieved by minimizing a cost or energy function that represents the geometric and/or intensity differences between two images.[12–15] Matched point landmarks have also been used in nonrigid registration.[11, 16, 17] A transformation function determines the spatial relation between corresponding points in the image being registered. Given the coordinates of a point in the reference image, a transformation function finds the coordinates of the same physical point in the target image. A transformation function makes it possible to create an image flow, showing the local deformation or motion of the target image with respect to the reference image.

2.1. Landmark Selection and Correspondence

Not all points in an image carry the same amount of information. Point landmarks carry more information than other points. Depending on the problem under consideration, different properties may be sought of the point landmarks. In image registration, the main role of point landmarks is to facilitate correspondence between images. For the correspondences to be accurate, it is required that the landmarks be taken from locally unique and informative. To accurately describe local geometric differences between two images, it is required that the landmarks be spread over the entire image domain rather than being concentrated in a few small neighborhoods. Landmarks that are very close to each other contain redundant information. Landmarks that are dispersed over an image contain less redundancy and are more suitable for image registration.

We consider point landmarks as highly informative and unique spherical image regions centered at edge points. Information is measured using entropy, and uniqueness is measured using auto-correlation. To ensure that the selected landmarks are widely dispersed, in addition to entropy and auto-correlation, distances between point landmarks are used when selecting them. The details of the proposed point landmark selection algorithm follow.

Algorithm: Point Landmark Selection (PLS)

1. Detect edges: Determine edges in the reference image by a 3-D edge detector.[11]
2. Locate informative points: From among the detected edges, keep only those that are locally most informative.
3. Identify unique points: From among the informative landmarks, keep only those that are centered at unique spherical templates of radius R.
4. Select N informative and unique landmarks: From among the detected point landmarks, keep only N that are widely dispersed over the image as follows:
   (a) Order the point landmarks in an *input* list according to their decreasing uniqueness measures.
   (b) Create an *output* list and let n represent the number of point landmarks in the output list. Therefore, initially $n = 0$ and the *output* list is empty.
   (c) If $N = n$ or if the *input* list is empty, *stop*. Otherwise continue.
   (d) Remove the point landmark from the top of the *input* list, append it to the *output* list, and increment n by 1.
   (e) Multiply uniqueness in the *input* list by $H = 1 - exp(-\frac{d_i^2}{D^2})$ and sort the list again, where d_i is the distance of the i^{th} landmark in *input* to the landmark just moved from *input* to *output*, and D is a parameter controlling the distance between selected landmarks. Then, go to Step 4c.

This algorithm has three parameters, R, s, and D. R is the radius of spherical templates used to calculate the informativeness and uniqueness of a selected landmark. In Step 1, image edges are determined. The standard deviation s of the Gaussian smoother used in edge detection should be selected from the knowledge about image noise and the required density of landmarks. If the image is noisy, or if a rather low density of landmarks is required, a larger standard deviation should be used than when the image is not noisy or when a rather dense set of landmarks is needed. D is used to revise the feature values of points remaining in the input list after one is removed. The larger the value for D, the farther away the influence of a selected landmark will go on remaining points in the input list, and the smaller the value for D, the more local this influence will be. A smaller D will allow landmarks that are near each other to be selected, while a larger D will increase the distance between adjacent landmarks, making the landmarks disperse more evenly over the image domain.

We used the zero-crossings of the Laplacian of Gaussian in 3-D as the edges. Since the zero-crossings represent locally maximum as well as locally minimum gradients and locally minimum gradients represent false edges, and also because the zero-crossings representing locally minimum gradients have small gradient magnitudes, from among the detected zero-crossings those that have small gradient magnitudes are removed. From among the remaining edges, a small percentage that have the highest uniqueness are kept and the rest are discarded. The portion of edges to be kept depends on the number N of landmarks requested.

Note that Step 2 keeps only informative local neighborhoods. An edge point is considered informative if the spherical area of radius R centered at it has a high entropy. The entropy is computed using the histogram of the spherical template:

$$e = -\sum_{i=0}^{255} p_i \log_2 p_i \tag{1}$$

where p_i is the probability that a randomly selected voxel has intensity i. It is more likely to find a correct correspondence using an informative template than a template that represents a rather uniform neighborhood. Also note that in addition to being informative, the points selected in Step 3 are centered at locally unique neighborhoods. Unique landmarks are more likely to produce correct correspondences than landmarks that are not locally unique. An edge point is considered unique if the spherical area of radius R centered at it produces a correlation peak when matched with spherical areas centered at adjacent voxels. The correlation is computed according to:

$$r = \frac{\sum_{i=0}^{M-1}(I_i^c - \overline{I^c})(I_i^n - \overline{I^n})}{\sqrt{\sum_{i=0}^{M-1}(I_i^c - \overline{I^c})^2}\sqrt{\sum_{i=0}^{M-1}(I_i^n - \overline{I^n})^2}} \tag{2}$$

where M is the number of points in the template, I^c is the intensity of points from the template centered at the edge point, I^n is the intensity of points from the template centered at an adjacent voxel, and the overline represents the average intensity of a template.

The landmarks are selected in Step 4. After ordering the landmarks in descending order of their uniqueness values in Step 4a, an *output* list is created in Step 4b. n is used as the number of landmarks in the *output* list at any given time. Therefore, initially, it is 0. Step 4c checks to see whether there are landmarks left in *input* to process or if the desired number of landmarks is reached. If either test succeeds, the algorithm is stopped. Otherwise, the landmark from the top of the *input* list is selected and moved to the *output* list in Step 4d. Then, in Step 4e the uniqueness of points remaining in *input* are multiplied by H, a measure that is inversely proportional to the distance of the landmark just removed from *input* to landmarks remaining in *input*. Since H approaches 1 exponentially, it will affect only a small number of landmarks in the list that are near the point just removed. In fact, only if a point in the list is within distance $5D$ of the point just removed, its uniqueness value is revised and it is repositioned in the list. Otherwise, the uniqueness value and the position of the point in the list is not changed. Step 4e ensures that landmarks that are very close to the landmarks already selected are not picked and preference is given to landmarks in image regions where no landmark in the neighborhood has been selected.

In image registration, coordinates of corresponding point landmarks are needed to determine the transformation function. To find the correspondences, templates are selected centered at landmarks in the reference image and searched for in the target image. The centers of matched templates in the images are then taken as corresponding landmarks. We select landmarks in the reference image according to Algorithm PLS and then determine their correspondences in the target image by template matching as follows:

Algorithm: Point Landmark Correspondence (PLC)

1. Select N landmarks in the reference image: Using Algorithm PLS, select N unique, highly informative, and well-dispersed landmarks in the reference image and save the selected points in list *input*.
2. Find corresponding landmarks in the target image: This involves the following:
   (a) *Initialize:* Let n show the number of correspondences found so far. So, initially, $n = 0$. Create list *output*, which will contain the coordinates of corresponding landmarks in the images. Initially this list is empty.
   (b) *Take the next landmark from input list:* If $n = N$, stop. Otherwise, take the next landmark from *input*, (x_n, y_n, z_n), and increment n by 1.
   (c) *Find the corresponding landmark in the target image:* Take a spherical template centered at (x_n, y_n, z_n) in the reference image and search for it in a spherical neighborhood of radius r centered at (x_n, y_n, z_n) in the target image. Determine the window in the target image that is most similar to the template. Let the coordinates of the center of the best-match window be (X_n, Y_n, Z_n).
   (d) *Determine the coordinates with sub-voxel accuracy:* Using the similarities obtained within the neighborhood of (X_n, Y_n, Z_n), determine the best-match position with sub-voxel accuracy by fitting a tri-quadratic function to the similarities and locating the function's peak in 3-D. Assuming the coordinates of the peak are (X'_n, Y'_n, Z'_n), append $[(x_n, y_n, z_n), (X'_n, Y'_n, Z'_n)]$ to list *output*. Then, go to Step 2*b*.

The similarity between a template and a window is determined by finding their mutual information. Mutual information is computed as follows:

$$I(T, W) = \sum_a \sum_b P_{TW}(a, b) \log \frac{P_{TW}(a, b)}{P_T(a) P_W(b)} \tag{3}$$

where $P_T(a)$ is the probability that the intensity at an arbitrary voxel in T is a, $P_W(b)$ is the probability that the intensity at an arbitrary voxel in W is b, and $P_{TW}(a, b)$ is the probability that intensity a in T lies on top of intensity b in W. If the template and the window contain similar patterns, their intensities will be highly dependent and they will produce a high mutual information. However, if the template and the window do not contain similar patterns, they will produce a small mutual information. Therefore, the higher the value of the mutual information, the more similar the template and the window will be. Instead of examining the entire search area in the target image to find the best-match position of a template, search is performed only at points in a small neighborhood surrounding the landmark in the target image. This means that only windows centered at the landmark and at voxels surrounding it in the target image are examined to find the window that best matches the template in the reference image.

To determine the best-match position with sub-voxel accuracy, in Step 2d a tri-quadratic function is fitted to the similarities obtained within the neighborhood centered at the best-match position obtained in Step 2c by the least-squares method. A tri-quadratic function is defined by

$$F(X, Y, Z) = aX^2 + bY^2 + cZ^2 + dXY + eXZ + fYZ + gX + hY + iZ + j. \tag{4}$$

Parameters a–j are determined by minimizing the root-mean-squared error in fitting function (4) to the coordinates and similarity of the best-match at (X_n, Y_n, Z_n) and those of 26 voxel surrounding it. The peak of this function is then located with sub-voxel accuracy and used as the point corresponding to (x_n, y_n, z_n).

The search area size (parameter r) should be chosen using information about the maximum brain shift. r should be selected large enough not to miss the correspondences. In order to reduce r without missing the correspondences, the images may be rigidly registered first using a small number of correspondences.[11] This will remove global translational and rotational differences between the images. If the images are already globally aligned, parameter r will depend on the maximum local geometric difference between the images.

2.2. Transformation Function

Knowing a set of corresponding point landmarks in two images, many transformation functions can be found to map the point landmarks to each other. The right transformation function will also map the remaining points in the images. Some transformation functions, although mapping corresponding point landmarks to each other accurately, they deform the target image too much in areas away from the landmarks, causing large errors in registration. Since a transformation is computed from the landmark correspondences, error in the correspondences carry over to the transformation function. It is desired that a transformation smooth the noise and small inaccuracies in the correspondences. A transformation function must account for the local geometric differences between the images being registered.

Suppose two volumetric images with N corresponding landmarks are available. Let $(x_i, y_i, z_i), i = 1, \ldots, N$ represent the coordinates of the landmarks in the reference volume, and $(X_i, Y_i, Z_i), i = 1, \ldots, N$, represent the coordinates of the corresponding landmarks in the target volume. The transformation that registers the images has three components f, g, and h, satisfying

$$X_i \approx f(x_i, y_i, z_i), \tag{5}$$
$$Y_i \approx g(x_i, y_i, z_i), \qquad i = 1, \ldots, N. \tag{6}$$
$$Z_i \approx h(x_i, y_i, z_i), \tag{7}$$

By substituting the coordinates of a point from the reference volume in function f, the corresponding X coordinate for the same point in the target volume is obtained. Similarly, by substituting the coordinates of a point from the reference image in functions g and h, the corresponding Y and Z coordinates for the same point in the target image are obtained.

Each component of a transformation function can be considered a function approximating scattered data in 3-D:

$$(x_i, y_i, z_i, X_i), \tag{8}$$
$$(x_i, y_i, z_i, Y_i), \qquad i = 1, \ldots, N. \tag{9}$$
$$(x_i, y_i, z_i, Z_i), \tag{10}$$

Therefore, the transformation function that deforms the target image to spatially align with the reference image can be viewed as three volume functions fitting data at scattered 3-D points. With that in mind, the problem of determining the transformation function becomes equivalent to the well-researched problem of multidimensional data approximation.

Given a set of irregularly spaced data in 3-D:

$$(x_i, y_i, z_i, f_i) : i = 1, \ldots, N, \tag{11}$$

we would like to find a function f that approximates the data anywhere in the xyz domain. It is generally required that function f have no overshoots or fluctuations. We propose an algorithm that uses rational Gaussian blending functions with linear weights. Estimating the weight functions from the data in a small neighborhood ensures that local details in data are reproduced well, without creating unwanted overshoots and fluctuations.

In addition, extending all weight functions over the entire approximation domain covers holes in data. Using blending functions with widths inversely proportional to local density of data automatically adjusts the surface to local details.

To approximate irregularly spaced data in 3-D, we use

$$f(x,y,z) = \sum_{i=1}^{N} W_i(x,y,z) l_i(x,y,z), \tag{12}$$

where $W_i(x,y,z)$ represents the influence of local polynomial $l_i(x,y,z)$ on the approximating function at point (x,y,z). The weight functions are selected such that their sum is 1 everywhere in the approximation domain. The purpose of the weight functions is to ensure that a functional value at a point is obtained mostly from data in the neighborhood of the point. The width of the weight function determines the level of detail in approximation. Narrow weight functions enhance local details, while wide weight functions average out local details. Each data point has an associating polynomial. The polynomial provides information about the gradient the data in the neighborhood of the data point and determines the local shape of the approximation function.

The polynomial of degree one describing the gradient of the function in the neighborhood of the i^{th} data point is represented by

$$l_i(x,y,z) = a_i x + b_i y + c_i z + d_i \tag{13}$$

where the coefficients a_i, b_i, c_i, and d_i of the i^{th} polynomial are determined by fitting a linear volume to data point i and at least two other points in that neighborhood. Parameter d_i is then readjusted by making the volume evaluate to f_i at (x_i, y_i, z_i), or

$$f(x_i, y_i, z_i) = f_i. \tag{14}$$

The size and shape of the neighborhood to be used will depend on the density and organization of the data. The main objective in finding the local polynomial is to capture local data gradient information.

The coefficients of the ith local polynomials are calculated using data within a sphere around the i^{th} point, containing a fixed number of points, n. When all n data points in a neighborhood lie on a plane, n is increased until the points no longer lie on the plane. In a neighborhood where the density of data is high, a small spherical area is sufficient to contain the n points, while in a neighborhood where the density of data is low, a larger spherical area is needed. The radii of the spherical areas containing the n data points are used as the widths of the weight functions. In this manner, the width of the weight function associated with a local polynomial is adjusted to the density of data in that neighborhood.

The weight functions define the degree of influence of different polynomials on a surface point. Narrow weighting functions force the approximating function to follow local details in data, while wide weight functions smooth local details. In our algorithm, rational Gaussian weight functions are used. By changing the standard deviations of the Gaussians, the widths of the weight functions can be controlled. The sum of the Gaussian weight functions is 1 everywhere in the approximation domain as defined by

$$W_i(x,y,z) = \frac{G_i(x,y,z)}{\sum_{k=1}^{N} G_k(x,y,z)} \tag{15}$$

where $G_i(x,y,z) = \frac{1}{\sigma_i} exp\{[(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2]/2\sigma_i^2\}$. The standard deviations of the Gaussians determine the widths of the weight functions. We let the standard deviations of the Gaussians be equal to the radii of the spherical areas containing n data points. Note that since the radii of the local neighborhood depend on n, as n is increased, the local neighborhoods widen. This makes it possible to control the smoothness/detailedness of the approximation by changing n.

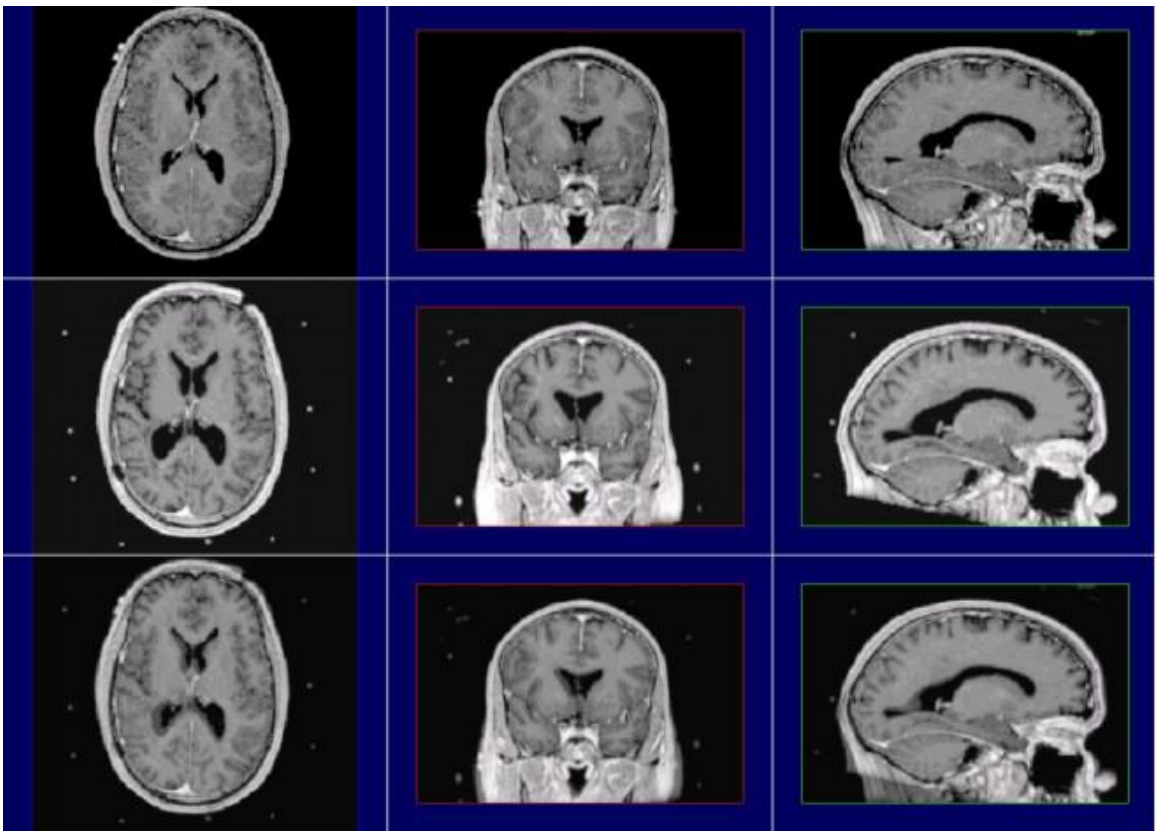

Figure 1. The first two rows show corresponding orthogonal cross-sections of two MR images of the same patient after rigid registration, and the third row shows corresponding cross-sections when overlaid.

3. RESULTS

3.1. Accuracy

To determine the accuracy of the proposed flow visualization algorithm the images shown in Figure 1 were used. A sub-volume of interest was selected in the rigidly registered images. The sub-volume in the target image was deformed with a known nonlinear transformation function defined by

$$\alpha exp\{-\frac{(x-x_i)^2+(y-y_i)^2+(z-z_i)^2}{2\sigma^2}\} \tag{16}$$

where α is the maximum deformation or brain shift that is believed to occur during a surgery, (x, y, z) is the center of the sub-volume, and σ is the inverse rate with which the deformation fades away with respect to the center point. The smaller the σ, the higher the rate at which the deformation fades away. By changing parameters α and σ, different target volumes are obtained. Since the amount of deformation between the images to be registered was known in advance, the root-mean-squared (RMS) and the maximum (MAX) errors between the true and the estimated shifts were computed and used to measure the accuracy of the proposed algorithm. The accuracy results are summarized in Table 1. All numbers are in millimeters.

3.2. Qualitative Assessment of Brain Shift

The left column in Figure 2 depicts rigid registration of MR brain volumes of the same patient who underwent surgery to remove a brain tumor. The images after nonrigid registration are depicted in the middle column. Nonrigid registration represents deformations near the ventricles that have been missed by rigid registration. To

Table 1. RMS and MAX approximation errors in millimeters as a function of σ. Parameter α was equal to σ.

	$\sigma = 2$	$\sigma = 4$	$\sigma = 6$	$\sigma = 8$	$\sigma = 10$	$\sigma = 12$
RMS	0.0185	0.1060	0.2931	0.4875	0.5400	0.6470
MAX	0.2959	1.3797	1.9919	3.3398	3.2241	3.2399

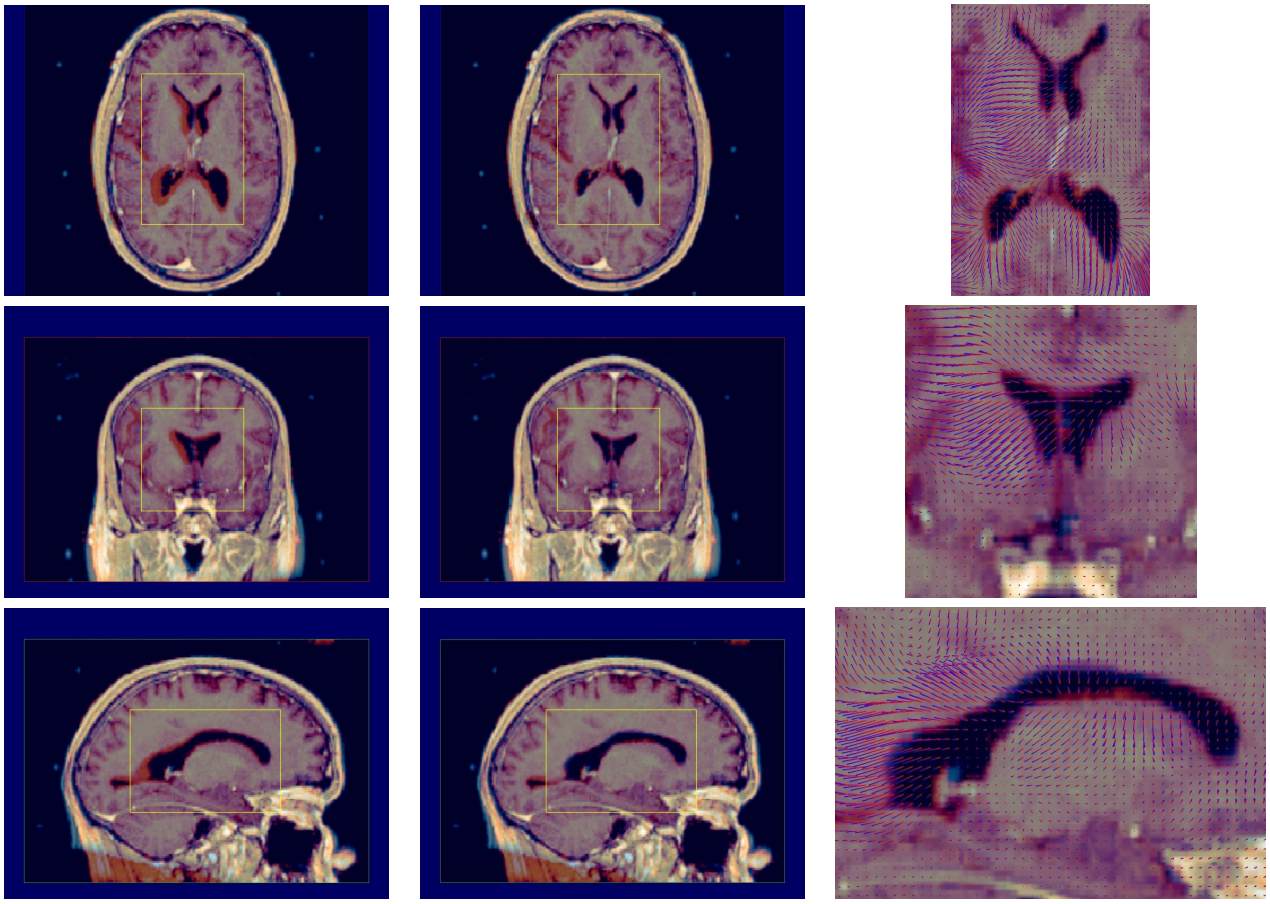

Figure 2. (left) Orthogonal cross-sections of overlaid images after rigid registration. (middle) The same cross-sections after nonrigid registration. (right) Vector flow depicting the magnitude and direction of brain shift.

visualize the local motion of brain tissues due to surgery, the proposed flow visualization algorithm was applied to the highlighted sub-volumes of interes in Figure 2. Performing a nonrigid registration with 10,000 corresponding point landmarks and using the proposed transformation function estimated the local motion of brain tissues, deforming the target volume to the geometry of the reference volume. Since the transformation function contains information about the displacement of the individual voxels in the target volume with respect to the reference volume, it can be used to create an image flow representing brain deformation. One end of a flow line shows a voxel in the target volume, and the other end shows the corresponding voxel in the reference volume. The image flow created in this way depicts brain shift caused by removal of the tumor.

The algorithm was also tested visualizing the local motion of brain tissues due to the growth of tumors. Figure 3 shows an example where two rigidly registered MR brain volumes of a patient taken at different times. The orthogonal cross-sections of the volumes are displayed in the top and middle rows. The deformation is quite obvious near the ventricles and remains unaccounted for after rigid registration. The bottom row shows

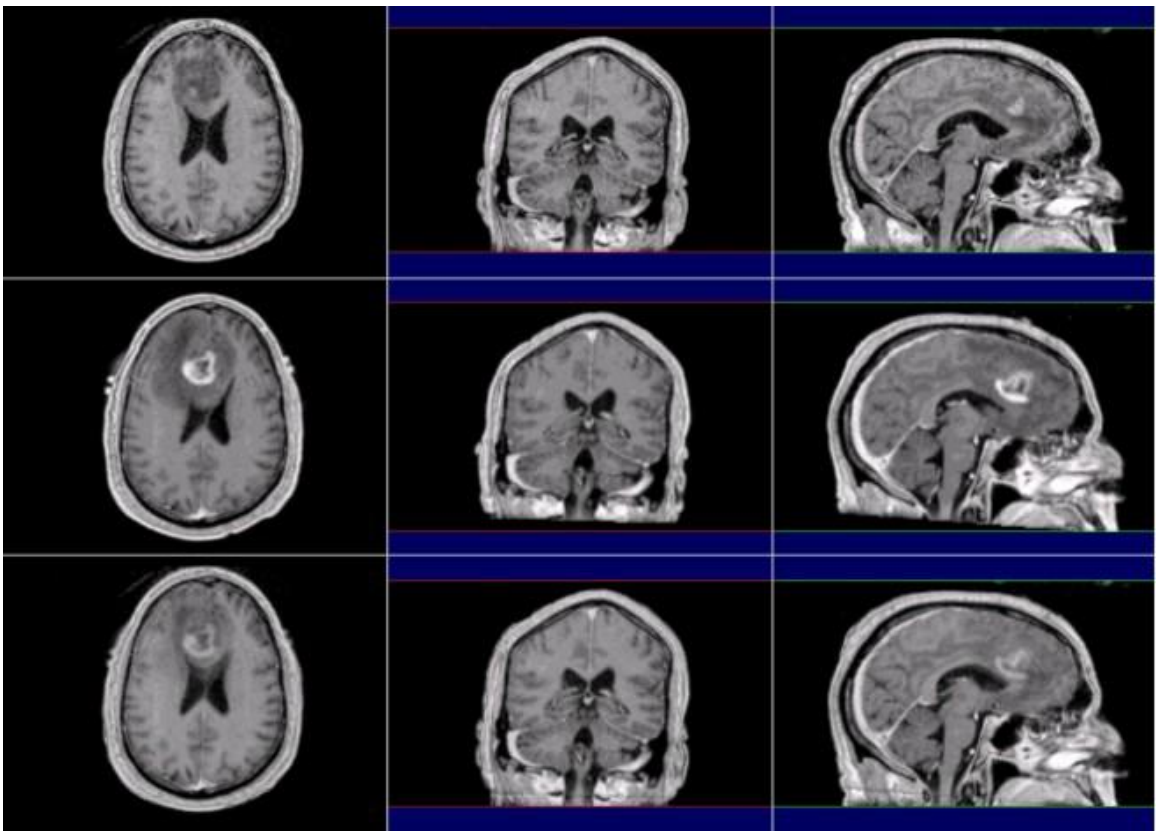

Figure 3. Rigid registration of MR brain images of a patient taken before and after a surgery. Corresponding orthogonal cross-sections in the image volumes are shown in rows 1 and 2. Row 3 shows overlaying of the corresponding cross-sections.

the overlaid reference and target volumes after the rigid registration. To account for brain deformations and to visualize the local motion of brain tissues due to the growth of the tumor, the proposed flow visualization algorithm was applied to the ventricular area within the highlighted sub-volumes as shown shown in Figure 4. The estimated transformation has been able to warp the target volume to the geometry of the reference volume as is evidenced by the overlaid cross-sections. The image flow shows the amount and direction of warping applied to the target geometry to obtain the reference geometry. The image flow also shows the amount and direction of brain shift occurred between the times the images were obtained.

4. CONCLUSIONS

A new algorithm for visualizing image flow in time-series MR or CT brain images acquired of the same patient was described. The method first establishes correspondence between point landmarks in two image volumes and then from the correspondences computes a transformation function that maps points in one image to points in another image. The computed transformation function does not require the solution of a system of equations and has the ability to adjust itself to various types of geometric differences between images. The obtained transformation function contains information about the local motion of one image with respect to another and can be used to create an image flow that represents the magnitude and direction of local brain shift.

The utility of the proposed method was evaluated using synthetic as well as real image volumes. Experimental results show that the method is able to detect and quantify brain shift. The developed flow visualization can convey to the surgeon the information needed to assess brain shift during or after a surgery. The image flow shows the amount and direction of brain shift. The process of computing and visualizing an image flow takes

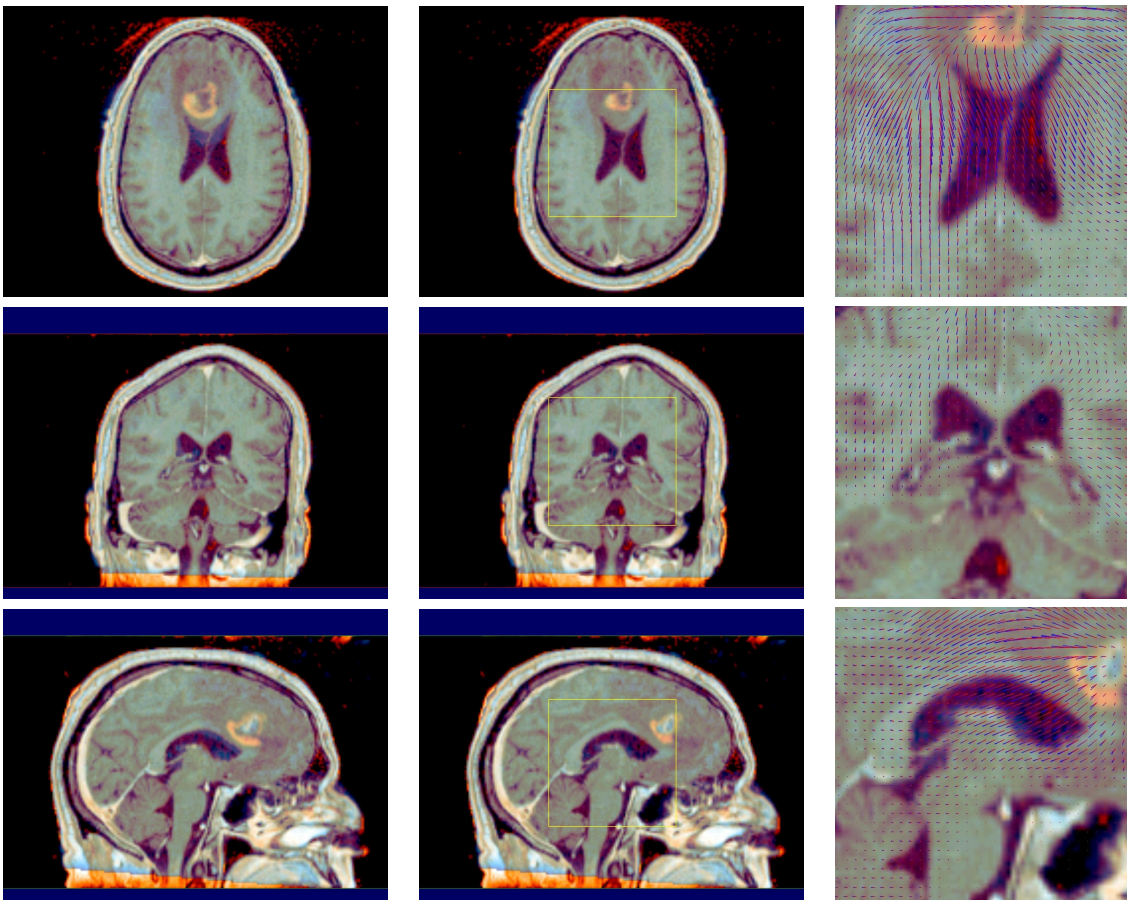

Figure 4. (left) Overlaid corresponding cross-sections of images after rigid registration. (middle) Overlaid corresponding cross-sections after nonrigid registration. (right) Image flow quantifying brain shift.

from a few to several minutes depending on the image size and the number of landmarks used to register the images.

REFERENCES

1. R. D. Bucholz, "The correction of stereotactic inaccuracy casused by brain shift using intraoperative ultrasound device," *Lecture Notes in Computer Science* , pp. 459–466, 1997.
2. J. Koivunkangas, "Ultra-sound controlled neuronavigator-guided brain surgery," *J. Neurosurg.* **79**, pp. 36–42, 1993.
3. H. Dickhaus, "Quantification of brain shift effects by MR-imaging," *Proc. 19th Int'l Conf. IEEE Engineering in Medicine and Biology* **19**, pp. 491–494, 1997.
4. D. W. Roberts, "Intraoperative brain shift and deformation: A quantitative analysis of cortical displacement in 28 cases," *Neurosurgery* **43(4)**, pp. 749–760, 1998.
5. M. Merickel, "3-D reconstruction: The registration problem," *Computer Vision, Graphics and Image Processing* **42(2)**, pp. 206–219, 1988.
6. A. Goshtasby, "Registration of images with geometric distortion," *IEEE Transactions on Geoscience and Remote Sensing* **26(1)**, pp. 60–64, 1988.
7. F. L. Bookstein, "Principal warps: Thin plate splines and the decomposition of deformations," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11(6)**, pp. 567–585, 1989.
8. R. Szeliski and S. Lavallee, "Matching 3-D anatomical surfaces with nonrigid deformations using octree splines," *Intl. Journal Computer Vision* **8**, pp. 171–186, 1996.
9. R. L. Hardy, "Theory and applications of the multiquadric-biharmonic method," *Computers Math Applic* **19(8/9)**, pp. 163–208, 1990.
10. D. Ruprecht and H. Muller, "Image warping with scattered data interpolation," *IEEE Computer Graphics and Applications* **15(2)**, pp. 37–43, 1995.
11. A. Goshtasby, *2-D and 3-D Image Registration: for Medical, Remote Sensing, and Industrial Applications*, Wiley Press, April 2005.
12. R. Bajcsy and C. Broit, "Matching of deformed images," *Proc. Int'l Conf. Pattern Recognition* , pp. 351–353, 1982.
13. D. Terzopoulos, A. Witkin, and M. Kass, "Constraints on deformable models: Recovering 3-D shape and nonrigid motion," *Artificial Intelligence* **36(1)**, pp. 91–123, 1988.
14. P. Thompson and A. Toga, "A surface-based techinque for warping 3-dimensional images of the brain," *IEEE Trans. Medical Imaging* **15(4)**, pp. 402–417, 1996.
15. C. Devatzikos, "Spatial transformation and registration of brain images using elastically deformable models," *Computer Vision and Image Understanding* **66(2)**, p. 207, 1997.
16. A. Goshtasby, "Piecewise cubic mapping functions for image registration," *Pattern Recognition* **20(5)**, pp. 525–33, 1987.
17. A. Goshtasby, "Image registration by local approximation methods," *Image and Vision Computing* **4(6)**, pp. 255–61, 1988.