

Fully Automatic Elastic Registration of MR Images With Statistical Feature Extraction

Martin Held*

Werner Weiser

Franz Wilhelmstötter

Universität Salzburg

Institut für Computerwissenschaften

A-5020 Salzburg, Austria

[held,wweiser,fwilhelm]@cosy.sbg.ac.at

Abstract

We present a fully automatic scheme for the registration of MR images. The registration is carried out as a combination of an affine and an elastic transformation. The affine part is generated by means of an affine Principal Components Analysis (PCA), which is an extension of the standard rigid PCA. We use the affine PCA as a preparatory step to guarantee maximum spatial similarity for the subsequent elastic transformation. The elastic transformation itself is based on a displacement vector field generated by means of Monte Carlo methods. Contrary to other Monte Carlo methods that define feature boundaries based on the grey-value transition of adjacent pixels, we make use of more accurate feature boundaries segmented by means of statistical feature extraction methods. We also present a validation method for verifying the segmentation results for simulated MR images. Although discussed in the context of medical imaging, our approach can also be applied as a general-purpose registration method in other fields of image processing. We conclude this paper with a discussion of the results obtained.

Keywords

Registration, Segmentation, Monte Carlo Warping, PCA, Medical Imaging, k-means Algorithm, Maximum a-posteriori Algorithm.

1 Introduction

Medical imaging plays a major role in the modern medical diagnostic. The recent improvements of the image acquisition techniques has caused a tremendous increase of raw digital image data, making it imperative to provide humans with tools for the fully automatic handling and evaluation of the data.

Since different imaging tasks require different acquisition techniques, it is common for medical imaging to deal with images of the same object that were derived from different sources. Similarly, the images may differ in acquisition time or viewpoint. Rather than viewing two images of the same object side by

side, it is often preferred to superimpose them such that features¹ common to both images match. This task of transforming one image such that it can be superimposed by another image is known as image registration. For background readings on image registration, reference is given to the surveys by van den Elsen et al. [vdEPV93] and Hajnal et al. [HHJ01].

In order to simplify the registration task a feature extraction step can be applied as a preprocessing. Feature extraction also is a widely studied topic in image processing; see [MST94] for a survey and introduction.

More formally, the image registration process deals with two images A and B . Image B denotes the target image and image A is the original image that should be transformed into B . An image can be regarded as a function that assigns each member of the image domain one value out of a fixed set of shades of grey (or colors). Typically, the domain will consist of pixels placed on a regular grid in two or three dimensions,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

¹In terms of medical image processing, “feature” means tissue type like bone, grey or white matter.

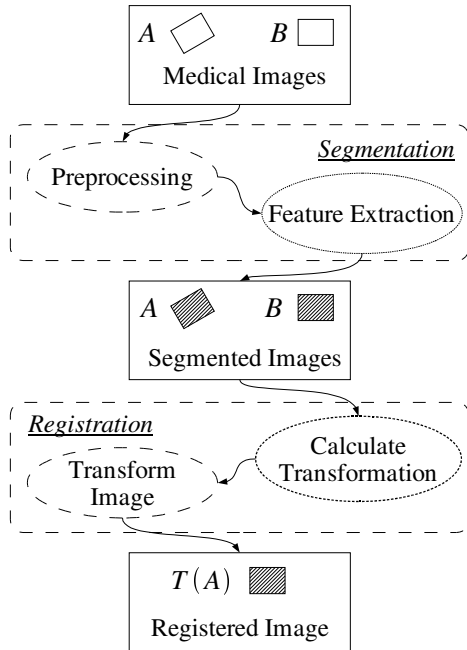


Figure 1: Flowchart of the registration process.

with integer coordinates.

The task of image registration is to find a transformation T such that $T(A)$ is as similar as possible to image B , where similarity is measured by a fitness function f . Diverse fitness functions can be used; a simple way to establish a fitness function is to count the number of pixels of B for which the color does not match the color of the corresponding pixel in $T(A)$. Thus, a transformation T is the better the smaller $f(T(A), B)$ is. In a more mathematical notation, the registration process can be formulated as the following optimization problem²:

$$T_{min} = \operatorname{argmin}_{T \in \mathbb{T}} \{f(T(A), B)\},$$

where \mathbb{T} denotes a class of transformations on the common image domain of A and B . Typical classes \mathbb{T} of transformations are rigid transformations, affine transformations, projective transformations and elastic transformations; see Foley et al. [FDF96] for an introduction to geometric transformations.

2 Overview

The main steps of a typical registration process are shown in the flowchart depicted in Figure 1. In this section we review how those steps are realized within our registration system.

²As usual, the operator argmin is defined as follows with respect to a function f and domain Ω of arguments to f : we have $\operatorname{argmin}_{\omega \in \Omega} = \omega'$ if $f(\omega') \leq f(\omega'')$ for all $\omega'' \in \Omega$.

Preprocessing: Depending on the type of image to be handled, it may be necessary to enhance the quality of the image by means of standard image processing techniques. Well-known preprocessing techniques include Gaussian filtering or median filtering for improving noisy images, edge enhancement, and adapting the image resolution.

Feature Extraction: The segmentation step is essential for our method because the knowledge of the feature boundaries makes the calculation of the transformation faster and more accurate. A large number of segmentation algorithms is known. However, several of these algorithms need additional user interaction. (E.g., a seed point is needed for region growing [AB94], and a start contour for snakes or ballons [MT00].) We apply statistical methods to guarantee a fully automatic segmentation process. We use the Maximum Likelihood Classifier for low-noise images and the Maximum a-posteriori Algorithm for noisy images [YK95]. Based on MR images generated by an MRI simulator [KEP99]³, we developed a validation method for both segmentation algorithms.

Calculate Transformation: Determining the transformation is the key step of a registration process. In our approach the transformation sought consists of two transformations applied consecutively. We use the first (affine) transformation to align the images in order to guarantee a maximum spatial similarity. This affine transformation is carried out by means of an affine PCA method, which is an extension of the rigid PCA described in [ABKC90]. The second (elastic) transformation is carried out by means of a displacement vector field. The calculation of the displacement vectors is done by Monte Carlo Warping [PSO+00].

Transform Image: Once these two partial transformations have been determined, the actual image registration is done by applying both partial transformations consecutively.

In the following section we give a short overview of the two segmentation methods studied and present details of the validation developed for the segmentation step. Section 4 starts with a short introduction to the specific type of elastic transformation that we use. The calculation of the two parts of the transformation (affine and elastic) is explained in Section 4.2 and Section 4.3. The results in Section 5, based on both, artificial test images and genuine medical MR images, show the power of our approach.

³McGill University's BrainWeb: Simulated Brain Database (SBD); URL: <http://www.bic.mni.mcgill.ca/brainweb/>.

This paper is accompanied by color images available on the WWW: point your browser to the URL given in [HWW03].

3 Feature Extraction and Validation

3.1 Feature Extraction

For the feature extraction step we use two different statistical classification methods: (a) The Maximum Likelihood Classifier (ML Classifier) and (b) the Maximum a-posteriori Segmentation Algorithm (MAP Algorithm) [YK95].

3.1.1 ML Classifier

An ML Classifier operates on the basis of the assumption that the grey values of the MR image were produced by a statistical process with a given probability distribution, where each feature $\omega \in \Omega$ has its own distribution. In the simplest case, a Gaussian distribution can be used.

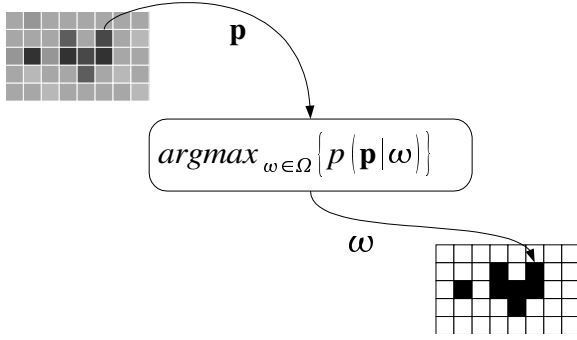


Figure 2: Schematic procedure of the ML Classifier

The main scheme of an ML Classifier is depicted in Figure 2. For each feature $\omega \in \Omega$ and every pixel \mathbf{p} in the MR image, the probability that \mathbf{p} belongs to ω , denoted by $p(\mathbf{p}|\omega)$, is calculated and the feature with the highest a-priori probability is assigned to \mathbf{p} in the segmented image. (This image is also known as feature image.)

3.1.2 MAP Algorithm

One drawback of an ML Classifier is that it does not care about the color of neighboring pixels. Therefore, noisy MR images tend to be segmented into noisy feature images. A MAP Algorithm handles this problem by introducing a Markov field or Gibbs field [Li01] into the segmentation method.

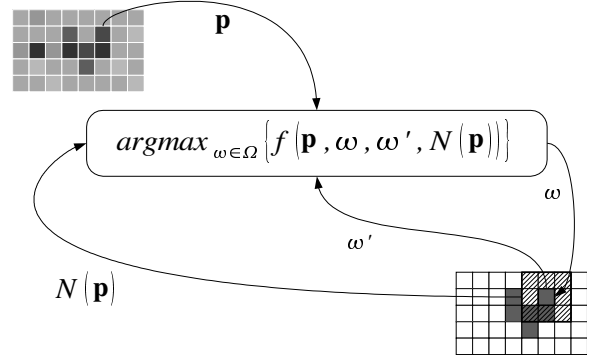


Figure 3: Schematic procedure of the MAP Algorithm

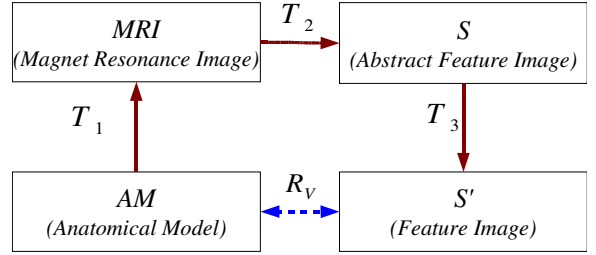


Figure 4: A-posteriori validation of a segmented image.

Figure 3 shows the iterative strategy of a MAP Algorithm. For every pixel \mathbf{p} in the MR image one considers a neighborhood $N(\mathbf{p})$ of \mathbf{p} . The function f uses the feature ω' computed during the previous iteration to calculate the current feature ω . Function f reflects the assumption that the pixels of $N(\mathbf{p})$ are more likely to belong to the same feature than to other features. This assumption is modeled by a Gibbs field with Gibbs potential β . The iteration terminates once consecutive feature images differ only in at most a given number of pixels.

3.2 Validation

The ML Classifier needs to choose only one parameter: the number of features k . In the MAP Algorithm the additional Gibbs potential β has to be chosen. Selecting a good parameter value is non-trivial for either algorithm. This subsection explains how we select appropriate parameters.

For the validation of the segmentation results we use simulated MR images. The MR images were generated from an anatomical model. This allows us to perform a pixel-by-pixel comparison between our segmentation result and the anatomical model. We take the number of incorrectly segmented pixels as a measure for the segmentation quality.

Figure 4 shows the flowchart for an a-posteriori vali-

dition. Transformation T_1 generates the simulated MR image from an anatomical model. This is done by the MRI simulator. Transformation T_2 stands for the feature extraction process and transformation T_3 is necessary to adapt the number of segmented features k of the segmented image to the number of tissues K of the anatomical model.

The transformation T_3 has to be chosen in a way that the number of incorrectly segmented pixels is minimized. This can be done by following the Maximum Likelihood principle [MST94], for $j \in \{0, \dots, k-1\}$:

$$T_3(j) := \operatorname{argmax}_{i \in \{0, \dots, K-1\}} \{p(j|i)\},$$

where $p(j|i)$ denotes the probability that some pixel \mathbf{p} in the feature image is segmented to feature j if the same pixel in the anatomical model belongs to feature i . With

$$p(j|i) = \frac{p(j \cap i)}{p(i)}$$

we get

$$T_3(j) := \operatorname{argmax}_{i \in \{0, \dots, K-1\}} \left\{ \frac{p(j \cap i)}{p(i)} \right\},$$

where $p(j \cap i)$ denotes the probability that tissue i of the anatomical model is segmented as the feature j by the segmentation algorithm. For T_3 we do not have to calculate the probabilities $p(j \cap i)$ and $p(i)$ explicitly. It is sufficient to calculate the absolute frequency matrix H with K rows and k columns. The single values $h_{i,j}$ of H specify how often the tissue i is segmented to the feature j (for the same coordinates). Since

$$h_{i,j} \propto p(j \cap i),$$

and

$$\sum_{j=0}^{k-1} h_{i,j} \propto p(i),$$

we obtain

$$T_3(j) := \operatorname{argmax}_{i \in \{0, \dots, K-1\}} \left\{ \frac{h_{i,j}}{\sum_{j=0}^{k-1} h_{i,j}} \right\}.$$

3.2.1 Validation of the ML Clusterer

For the ML Clusterer we want to optimize the number of segmented features k . Therefore we have to vary k . For each k we determine T_3 and then we count the number of incorrectly segmented pixels. Experiments showed that for the MR images used a value of $k = 6$ leads to a minimum of incorrectly segmented pixels, for a wide range of noise levels ranging from 3% to 9%. (Experiments with other MR data sets of human heads convinced us that picking a value out of the interval 4–7 is a good choice for k for most data sets.)

3.2.2 Validation of the MAP Algorithm

The validation of the MAP Algorithm is not done directly on the anatomical model respectively on the segmented feature image. We first calculate the variance image of the anatomical model and the segmented feature image and then we compare these variance images with the help of mutual information (MI). The calculation of the variance images "amplifies" the noise of a noisy segmentation.

For calculating the variance image we use the N_{26} neighborhood of a pixel \mathbf{p} . (Recall that MR images constitute three-dimensional data sets.) The mean value and variance are computed using standard formulas; we only have to make sure that the variance image is independent of the feature numbering. The mutual information between two images is calculated by means of the absolute frequency matrix H which we used for the calculation of the transformation T_3 .

We tuned the Gibbs potential β based on the information on the mutual information between (the variance images of) the anatomical model and the segmented feature image. The number of segmented features k was set to 6. The value of β depends on the quality of the images. We ended up using $\beta := 0.01$ for clear images (3% noise) and $\beta := 0.05$ for noisy images with 9% noise. (If an MR scanner does not provide information on the noise of the images generated then an estimate of the noise can be computed as explained in [SdDV⁺98].)

4 Registration

The primary purpose of the registration is to compensate or minimize geometric distortions between two or more images. This task can be handled by computing and executing different kinds of transformations.

4.1 Transformations

We used elastic transformations [Boo92] for the Monte Carlo Warping described in Section 4.3. Elastic transformations offer the highest amount of flexibility. In the most general case it is possible that every pixel is transformed in its own very specific way. Therefore elastic transformations can not be represented by a standard 4×4 homogeneous matrix. Common representations include linear combinations of basis functions (e.g., thin plate splines by Bookstein [Boo92]), fluid continuum models (based on the Navier-Stokes equation) [CRM96], and an optical flow model [HS81].

Since Monte Carlo Warping yields displacement vectors, we used a displacement vector field \mathbf{V} to en-

code an elastic transformation:

$$\mathbf{V} := \{(\mathbf{p}_i, \mathbf{q}_i) : 0 \leq i < M\},$$

where \mathbf{p}_i denotes the start point and \mathbf{q}_i denotes the end point of the i -th vector \mathbf{v}_i . To transform a single point \mathbf{p} , the contributions of all displacement vectors have to be summed. The weight (*extent*) of the contribution of one displacement vector \mathbf{v}_i depends on the distance between the point \mathbf{p} and the start point \mathbf{p}_i of \mathbf{v}_i . The distance is usually rated linearly, quadratically or exponentially by a distance weighting function w_i for the i -th displacement vector. The example in the following equation uses the city-block distance with exponential rating: The distance weighting function w_i of a single point $\mathbf{p} = (x, y, z)$ and a start point $\mathbf{p}_i = (x_i, y_i, z_i)$ is given by

$$w_i(\mathbf{p}) = e^{-\|\mathbf{p}-\mathbf{p}_i\|_1} = e^{-(|x-x_i|+|y-y_i|+|z-z_i|)}.$$

This leads to an equation for an elastic transformation T for a point \mathbf{p} . Consider $\{\mathbf{p}_i : 0 \leq i \leq M-1\}$, which is a set of M reference points (“landmarks”) of the original image, and $\{\mathbf{q}_i : 0 \leq i \leq M-1\}$, which is the corresponding set of points for the reference image. Then $T(\mathbf{p})$ is obtained as

$$T(\mathbf{p}) := \mathbf{p} + \frac{\sum_{i=0}^{M-1} w_i(\mathbf{p})(\mathbf{q}_i - \mathbf{p}_i)}{\sum_{i=0}^{M-1} w_i(\mathbf{p})}.$$

This kind of transformation is also called warping.

4.2 PCA Method

The goal of a Principal Component Analysis [ABKC90] is to reduce the dimensions of a set $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$ of n vectors $\mathbf{p}_i \in \mathbb{R}^d$ such that more representative samples are obtained. The PCA method, which is also known as discrete Karhunen-Loeve transformation, replaces those n vectors with m -dimensional vectors, the so-called principal components, where $m \leq d$. For that purpose an orthogonal transformation of the d -dimensional vectors on an m -dimensional hyperplane is applied. This transformation is equivalent to solving the eigenvalue problem. As a result the principal components are the eigenvectors of the covariance matrix of the vectors, while the corresponding eigenvalues represent the variances in the directions of the principal axes. Figure 5 illustrates the concept just discussed.

For an application to our imaging problem, the PCA method can be described as follows:

1. Extract the n_A feature vectors $\mathbf{P}_A = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{n_A}\}$, describing a feature in image A and the n_B feature vectors $\mathbf{P}_B = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{n_B}\}$, describing the same feature in image B and compute the centers of gravity $S(\mathbf{P}_A)$ and $S(\mathbf{P}_B)$ of both corresponding features.

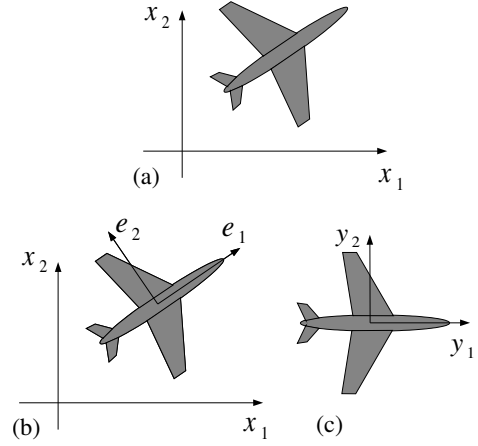


Figure 5: Principal Component Analysis. Sub image (a) shows a binary object in 2D. In sub image (b) the Principal Axes of the object are added and in (c) the object is aligned by means of the PCA method.

2. Carry out the PCA method for the feature considered in both images and compute a transformation for each of them.
3. Concatenate these two transformations to obtain the final transformation.

Generally the type of the resulting transformation is rigid (rPCA). We stretch the image along the principal axes by comparing the eigenvalues of both transformations. This stretching extends the rigid PCA to an affine PCA (aPCA), thus rendering the PCA method a better preprocessing step for the Monte Carlo Warping.

4.3 Monte Carlo Warping

We use displacement vector fields (described in Section 4.1) to model an accurate spatial correspondence between three dimensional medical data volumes. The method we describe in this section mostly follows the work of Pielot [PSO⁺00]. As mentioned in the previous subsection, displacement vector fields are encoded relative to a set of reference points (“landmarks”). As explained in the sequel, the selection of the reference points is based on Monte Carlo techniques (random walk) to find suitable points in the anatomical structure.

We start with superimposing a regular three-dimensional grid on both images. Every cell of the grid is treated as an individual sub-volume of the image. Inside every cell we employ a random walk to search for reference points. A reference point is found and added to a cell if the random walk detects a feature boundary. After the feature extraction a feature boundary is defined easily according to the feature difference

of two adjacent points. If no reference point was found after a user-defined maximum amount of steps⁴ of the random walk, this cell is declared to contain no reference point.

All reference points of a cell are represented by their arithmetic mean. If the corresponding cells in both images contain at least one reference point, a displacement vector can be built, and it is added to the displacement vector field. (The mean value of the reference points of the original image is the start point of the displacement vector.)

Recall that we adjust the images with the affine PCA method (see Section 4.2) prior to warping in order to guarantee maximum spatial similarity. We call this combined method PMC Warping (PCA Monte Carlo Warping).

4.4 Validation

To validate the result of the registration process suitable quality criteria are needed. In practice, the quality of a registration will often be checked visually by a user. To take the human out of the loop we need a quality measure which can be determined automatically by the computer. The measures we used are mutual information (MI), normalized mutual information (NMI), and cross correlation (CC). For all three measures the corresponding fitness function of the overlapping portion of the transformed original image $T(A)$ and the target image B has to be calculated. (It is obvious that only the overlapping portion of $T(A)$ and B can be used for the evaluation of the fitness function.)

5 Results

The two segmentation algorithms (ML Clusterer and MAP Algorithm) were tested on MR images produced by the MRI simulator, and on a variety of other MR data, with different levels of noise. For the later use by the Monte Carlo Warping smooth feature borders are desired. As evidenced by our experiments, for images with 3% noise the two segmentation algorithms produce mostly identical results. Since the ML Clusterer is 5–6 times more efficient than the MAP algorithm, it seems obvious to use the ML Clusterer for such data. However, as the amount of noise goes up the segmentation quality of the ML Clusterer goes down, and it becomes necessary to use the MAP Algorithm.

The registration method was first tested on artificial data with known results. The test images were generated to test specific properties of the registration

⁴If the grid contains a boundary then the probability that the random walk does not find this edge should be minimized. The probability can be estimated as the ratio of the number of boundary pixels over the number of overall pixels in the grid. In our implementation we allowed a maximum of 500 steps.

process. The registration process starts with the PCA for aligning the features. If the PCs of a feature are not unique⁵, the PCA can not find a correct transformation. Fortunately, in medical images such features rarely exists.

Figure 6 shows the power of Monte Carlo Warping for elastic transformations. However, Monte Carlo Warping will function best only if the spatial distortion between the features does not exceed a certain maximum. The maximum value permissible depends on the grid size, image size and feature size. The only parameter that can be adapted by the user is the grid size⁶. Note that the grid size can not be increased arbitrarily, since the assignment of the feature boundaries becomes ambiguous if more than one feature boundary cross a grid cell, which in turn may lead to inappropriate displacement vectors. The ability of the PCA to reduce spatial distortions among the features makes it an optimal preprocessing step for the Monte Carlo Warping. Therefore combining the PCA and the MC Warping generally generates better results than the individual methods.

Sample results for a real MR image of a human brain as target image are shown in Figure 7. The original image was created by a global affine transformation followed by a local elastic transformation of the target image. Table 1 lists the values of different fitness functions (MI , NMI and CC) for this sample registration task. It is easy to see that the registration results get better (higher values) as the flexibility of the transformation increases. Fairly similar results were obtained for other MR data sets. Due to lack of space additional images and test results are omitted; the reader is referred to [HWW03] for further material.

algorithm	MI	NMI	CC
“as is”	0.816	0.065	0.285
rPCA	1.152	0.092	0.526
aPCA	1.626	0.138	0.848
MC	0.821	0.075	0.292
PMC	1.634	0.138	0.855

Table 1: Experimental results for the registration task shown in Figure 7.

6 Conclusion

We present a new fully automatic elastic registration method for medical imaging which uses statistical fea-

⁵This is the case for squares, circles, cubes, spheres and so on.

⁶Since the Monte Carlo Warping is responsible for local deformations only, it is sufficient to divide each image dimension into 10 to 25 parts.

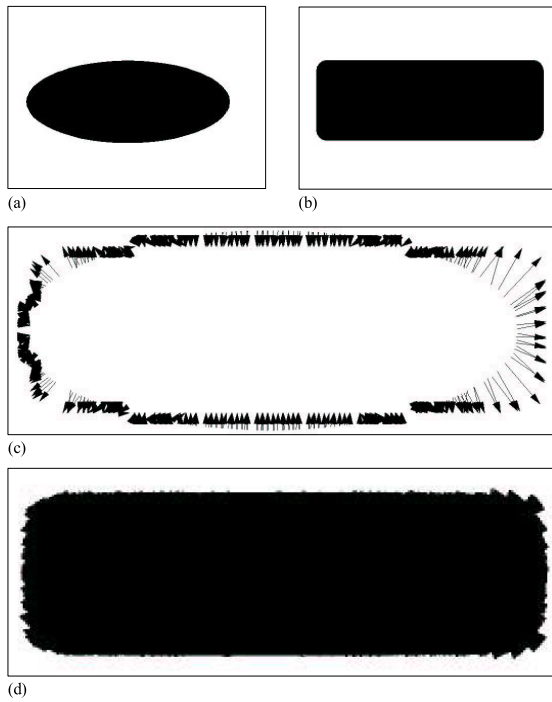


Figure 6: Power of the Monte Carlo Warping for elastic transformations. Sub figure (a) shows the original image, (b) shows the target image, (c) shows the displacement vector field created by the MC Warping algorithm and (d) shows the transformation of the original image.

ture extraction. In contrast to other registration methods that are also based on Monte Carlo methods we use an explicit segmentation step for boundary detection as a preprocessing for the actual registration. The quality of the feature extraction step directly influences the quality of the registration result. To increase the robustness of the overall method we use an affine PCA method to guarantee maximum spatial similarity before Monte Carlo Warping is applied.

As indicated by our experiments, the resulting PMC Warping yields better results than the individual methods. The generation of the displacement vector field is facilitated by the knowledge of accurate feature boundaries, which are computed during the segmentation preprocessing step.

References

[AB94] R. Adams and L. Bischof. Seeded Region Growing. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 16(6), June 1994.

[ABKC90] N.M. Alpert, J.F. Bradshaw, D. Kennedy, and J.A. Correia. The Principal Axes Transforma-

tion – A Method for Image Registration. *Journal of Nuclear Medicine*, 9:1717–1722, 1990.

[Boo92] F. Bookstein. Principal Warps: Thin-Plate Splines and the Decomposition of Deformation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):567 – 585, 1992.

[CRM96] G.E. Christensen, R.D. Rabbitt, and M.I. Miller. Deformable Templates using Large Deformation Kinematics. *IEEE Transaction on Image Processing*, 5(10):1435–1447, 1996.

[FDF96] J.D. Foley, A. Van Dam, and S.K. Feiner. *Computer Graphics in C. Principles and Practice*. Addison-Wesley, 2nd edition, 1996. ISBN: 0201848406.

[HHJ01] Joseph V. Hajnal, Derek L.G. Hill, and David J.Hawkes. *Medical Image Registration*. CRC Press, 2001.

[HS81] B. K. P. Horn and B. G. Schnuck. Determining optical flow. *Artificial Intellegence*, 17:185–203, 1981.

[HWW03] www.cosy.sbg.ac.at/~held/medimage/.../medimage.html, 2003.

[KEP99] R.K. Kwan, A.C. Evans, and G.B. Pike. MRI Simulation-Based Evaluation of Image-Processing and Classification Methods. *IEEE Transactions on Medical Imaging*, 18(11):1085–1097, November 1999.

[Li01] S.Z. Li. *Markov Random Field Modeling in Image Analysis*. Computer Science Workbench. Springer, 2001. ISBN 4431703098.

[MST94] D. Michie, D.J. Spiegelhalter, and C.C. Taylor, editors. *Machine Learning, Neural and Statistical Classification*. Prentice Hall, July 1994. ISBN: 013106360X.

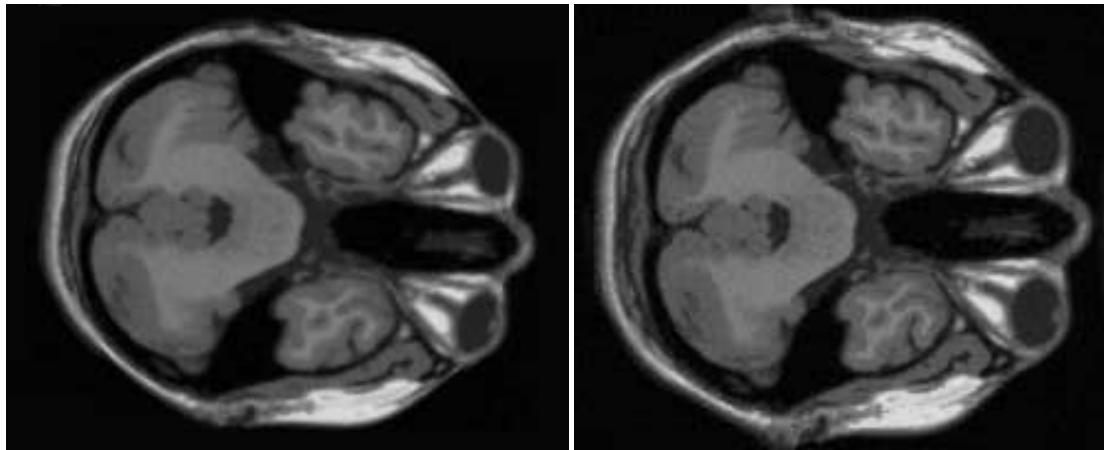
[MT00] T. McInerney and D. Terzopoulos. T-Snakes: Topology Adaptive Snakes. *Medical Image Analysis*, 4(2):73–91, 2000.

[PSO⁺00] R. Pielot, M. Scholz, K. Obermayer, E. Gundelfinger, and A. Hess. Point-Based Warping With Optimized Weighting Factors of Displacement Vectors. In K.M. Hanson, editor, *Medical Imaging 2000*, volume 3973, pages 1387 – 1395. 2000.

[SdDV⁺98] J. Sijbers, A.J. den Dekker, M. Verhoye, J. Van Audekerke, and D. Van Dyck. Estimation of Noise from Magnitude MR Images. *Magnetic Resonance Imaging*, 16(1):87–90, 1998.

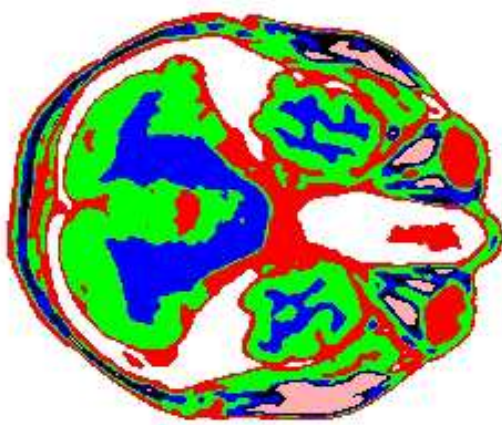
[vdEPV93] P.A. van den Elsen, E.J.D. Pol, and M.A. Viergever. Medical Image Matching - A Review with Classification. *IEEE Engineering in Medicine and Biology*, 12(1):26 – 39, 1993.

[YK95] M.X.H. Yan and J.S. Karp. An Adaptive Bayesian Approach to Three-Dimensional MR Brain Segmentation. U. of Pennsylvania, 1995. www.uphs.upenn.edu/bbl/software/.../files/segm-0.90.src.tgz.

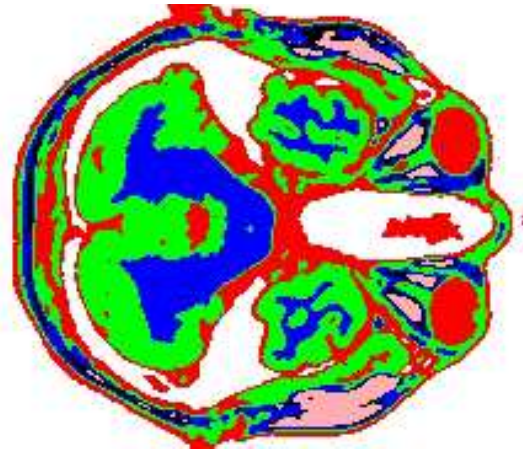


(a) Original image

(b) Target image



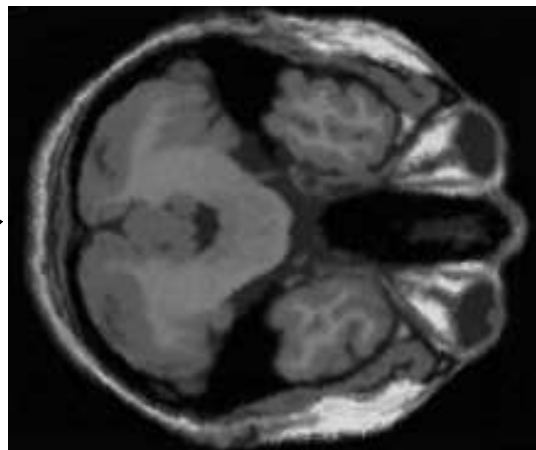
(c) Segmented original image



(d) Segmented target image



(e) Vector field



(f) Transformed original image

Figure 7: The first row shows the grey value images with 3% noise. The second row contains the segmentation result of the images above. Image (e) shows the displacement vector field created by the Monte Carlo Warping, and Image (f) shows the transformed original image (a).