

A Pyramid Approach to Subpixel Registration Based on Intensity

Philippe Thévenaz, Urs E. Ruttimann, and Michael Unser, *Senior Member, IEEE*

Abstract—We present an automatic subpixel registration algorithm that minimizes the mean square intensity difference between a reference and a test data set, which can be either images (two-dimensional) or volumes (three-dimensional). It uses an explicit spline representation of the images in conjunction with spline processing, and is based on a coarse-to-fine iterative strategy (pyramid approach). The minimization is performed according to a new variation (ML*) of the Marquardt–Levenberg algorithm for nonlinear least-square optimization. The geometric deformation model is a global three-dimensional (3-D) affine transformation that can be optionally restricted to rigid-body motion (rotation and translation), combined with isometric scaling. It also includes an optional adjustment of image contrast differences. We obtain excellent results for the registration of intramodality positron emission tomography (PET) and functional magnetic resonance imaging (fMRI) data. We conclude that the multiresolution refinement strategy is more robust than a comparable single-stage method, being less likely to be trapped into a false local optimum. In addition, our improved version of the Marquardt–Levenberg algorithm is faster.

Index Terms—Affine transformation, intramodal voxel-based registration, Marquardt–Levenberg nonlinear least-squares optimization, multiresolution, spline.

I. INTRODUCTION

THE NEED for image registration arises in many fields of research, most often in applications involving the comparison of a series of images. Of interest may be the detection of change, and the consolidation of data (or image fusion), where different images of the same object need to be brought into correspondence. Remote sensing and biomedical imaging are typical application areas. Due to the recurrent problem of registration, many solutions have been proposed [1]–[3].

Our specific interest is the detection of significant differences between biomedical images obtained from different subjects, or from the same subject at different times. In particular, this requires the three-dimensional (3-D) intramodal registration of positron emission tomography (PET) or functional magnetic resonance imaging (fMRI) images of the

brain. In both applications, the changes to be detected are very small (less than a tenth of the dynamic range for PET, and even less for fMRI), demanding accurate registration. This requirement rules out many of the methods involving finite search techniques or quantized parameter spaces; only subpixel methods are adequate. Furthermore, the difficulty of creating a robust, yet precise, algorithm for properly selecting landmarks in a 3-D grid rules out landmark-based registration, especially in PET images, since these are usually very noisy and cluttered with artifacts (see Fig. 1). Therefore, we prefer to explore methods that use the unaltered intensity of all image pixels, thus exploiting effectively all available information. Another aspect is the interpolation needed in applying the transformation. While the majority of published methods limit themselves to linear interpolation, we prefer a higher interpolation order to minimize image blurring and to achieve consistency in computing the spatial derivatives required for the registration process. A final need to be met by our registration method is versatility in selecting a deformation model: In some applications, such as intrasubject registration, a rigid-body transformation is appropriate, while in others, such as intersubject registration, a more complicated deformation model must be selected. Our deformation model considers the combination of a full 3-D affine transformation and an optional linear contrast change. We derive simpler models by restricting the affine transformation to specific parameter subsets, implementing combinations of rotation, translation and isometric scaling.

The Marquardt–Levenberg nonlinear optimization algorithm [4] is well suited for performing registration based on a least-squares criterion [5], [6]. In this paper, we introduce two extensions of this algorithm. First, we speed up its execution by taking advantage of the particular structure of our deformation model. Specifically, we reformulate the optimization problem in such a way that we can precompute most of the terms required for building the Hessian and the gradient of the criterion, instead of having to reevaluate them at every iteration as required in the traditional approach. As a second extension, we cast Marquardt–Levenberg into a multiresolution framework, using a coarse-to-fine iteration strategy, and propagating estimates for one level of a resolution pyramid from its prior level. Most iterations are carried out at the coarsest level, where the amount of data is so greatly reduced that the computational cost of one iteration is negligible. Once convergence has been reached at any particular level, a switch to a finer level is made, where

Manuscript received February 7, 1996; revised March 4, 1997. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Janusz Konrad.

P. Thévenaz and M. Unser are with the Swiss Federal Institute of Technology, CH-1015 Lausanne, Switzerland (e-mail: Phillippe.thevenaz@epfl.ch).

U. E. Ruttimann is with the National Institute on Alcohol Abuse and Alcoholism, National Institutes of Health, Bethesda, MD 20892-1256, USA (e-mail: ursr@nih.gov).

Publisher Item Identifier S 1057-7149(98)00537-5.

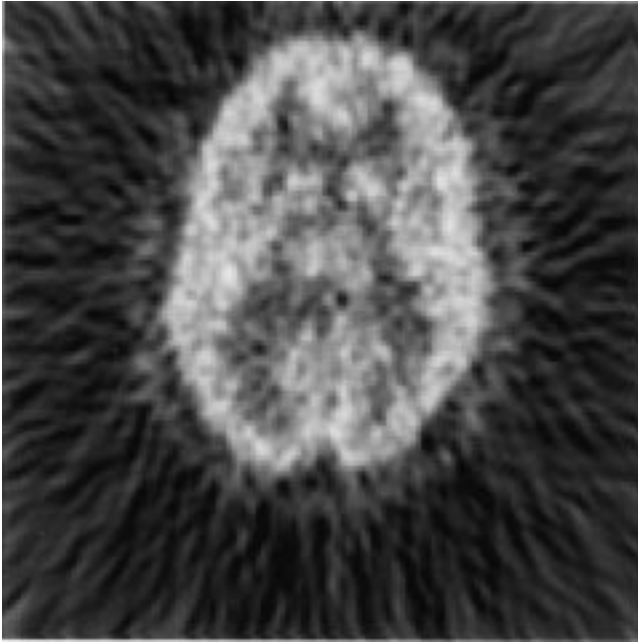


Fig. 1. Slice of a typical PET image.

only a few iterations are needed because of near-optimal initial conditions. For many types of optimizers, this strategy for convergence is significantly faster than a single-stage approach, but in the case of Marquardt–Levenberg (and our extension thereof), the benefits are even greater because this algorithm is superlinear and converges much faster than most other minimization schemes so long as the initial estimate is close to the correct solution. In addition, a multiresolution strategy improves robustness, in the sense that it decreases the likelihood of being trapped at a false local optimum.

A distinctive feature of our approach is that we consider spline interpolation models that are superior to those typically used for image registration (e.g., bilinear interpolation). By increasing the order of the spline, we can get arbitrarily close to the sinc interpolation model [7]. In practice, we use cubic splines because they are already remarkably close to this ideal, at a cost that is less than truncating and apodizing a sinc kernel [5]. Our spline model is well suited for computing image pyramids and for performing geometric transformations at various resolutions. It also allows for an easy computation of exact derivatives. By using the same model at each step, we ensure that the overall algorithm is internally consistent.

The algorithm that we propose is entirely automatic. Since it is pixel based, no landmarks are required. We have used it successfully without modification—not even tuning—for the registration of a whole variety of biomedical images including PET, MRI, and fMRI data, and high-resolution electron micrographs of virus particles.

A. Paper Outline

We begin with a short literature review in Section II. In Section III, we describe our registration procedure and include a rationale for the choice of our data-space, for the choice of its corresponding objective criterion, and for the

choice of two transformations: a general affine transformation and its restricted rigid-body version. In Section IV, we first give a descriptive computational approach to the standard Marquardt–Levenberg (ML) algorithm, and then explain how to achieve further acceleration by tuning it to our specific application, leading to a proposal for a new optimization algorithm (ML*). We also propose a heuristic for deciding when convergence has been reached. In Section V, we discuss the choice of the cubic spline interpolation model. In Section VI, we discuss the use of a multiresolution pyramid and its benefits. Finally, in Section VII, we provide several application examples before we conclude in Section VIII.

II. LITERATURE REVIEW

There are numerous ways to categorize the different image registration methods [1], [3], [8]. Here, we adopt the classification system of [1], which distinguishes three main categories, each of which can be further subdivided. The first category refers to the type of image feature considered, the second addresses the search space, or equivalently the type of transformation that is applied to the image, while the last describes the search strategy. Following this scheme, we give below a short discussion of the most important recent registration methods.

A. Image Features

The image features used in a given algorithm have important practical significance because they often determine what kind of images it is capable of registering. For example, spatial coordinates (landmarks) are well adapted to intermodal registration, where the purpose is to register two volumes measuring different properties of an object. However, the selection of landmarks is recognized to be a difficult problem [9], [10], whether done automatically [11] or manually [12]. For many images, this is a serious drawback because registration accuracy can be no better than what is achieved by the initial selection of landmarks. For practical reasons, the number and precision of landmark locations is usually limited. Hence, spatial coordinates and geometric primitives often oversimplify the data by being too sparse and imprecise.

By contrast, registration methods based on initial intensity values can make effective use of all data available [13]; if necessary, some binary masking or other weighting process may be introduced to emphasize special features. Robustness is controlled through the use of appropriate similarity measures [14]–[16]. With the use of theoretical models, intensity-based methods can produce continuous deformation fields [17].

B. Search Space

Geometric transformations can be divided into three categories: global, local, and displacement field. The first usually involves matrix algebra, in which a single (small) matrix characterizes the transformation of the entire image or volume; typical operations are translation, rotation, isotropic or anisotropic scaling, bi- or trilinear transformations, and quadratics [18]. The second category, sometimes called *elastic mapping*, allows the transformation parameters to exhibit

spatial dependence. These parameters are often defined on certain key points only, and interpolated on a region-by-region basis [19], [20]. Finally, true displacement fields, sometimes also called *optical flow*, result from the use of a (continuous) functional optimization scheme, in which an independent displacement is computed for each point in the image, with constraints arising from some *a priori* regularization [17], [21]–[23].

C. Search Strategy

Given a set of features and a parametric deformation, both the criterion to optimize and the optimization algorithm itself define the search strategy. The use of a least-squares criterion jointly with geometric primitives is popular [24], although it is sometimes replaced by more robust statistics, as those generated while using a distance map (chamfer transform) [25]. The least-squares criterion is also widely used with intensity values as image features [5], although researchers sometimes prefer more robust statistics, giving up maximum likelihood parameter estimation (in the presence of Gaussian white noise) in favor of assuring insensitivity to outliers [15]. The optimization algorithm reflects the choice of the criterion: When the latter is correlative in nature, gradient-based methods can be used. If the data are regularly spaced, such as in the case of pixel intensity values, both Fourier [26], [27] and wavelet approaches [17] are applicable. Several researchers have explored the possibilities of stochastic, or finite-search methods: dynamic programming [23], simulated annealing [28], or genetic algorithms [20]. Exhaustive search has also been investigated. With strong conditioning of the data, near real-time registration [29] can be achieved.

D. Problem Addressed

An alternative way to classify registration is to look at the type of problem it addresses. Again, three categories emerge: data fusion, motion estimation, and the detection of significant differences. In the first case, where noise reduction is often the goal, one takes advantage of the availability of multiple instances of supposedly identical data. In these instances, registration allows the extraction of common features, for example by averaging, or by more refined processes [30], [31]. Data fusion also arises when one needs to find the correspondence between two images acquired by two or more modalities; applications typically involve remote sensing [32] or medical imaging [33], [34]. In the context of motion estimation (second category), the problem is to estimate the displacement of a rigid object imaged on some background, with the added challenge of potential changes of pose [35]. Since typical applications are video coding, target tracking and autonomous vehicles, computational efficiency is an important aspect of registration methods used in this context. The third and last category encompasses the detection of significant differences, where the challenge arises because the object itself may change, aside from modifications of its orientation or its illumination. A new problem then appears, due to the fact that the registration process tries to align data that may be intrinsically dissimilar. This last consideration has sometimes

led to robust registration methods using an internal criterion insensitive to outliers [14]–[16]. After registration, the task usually proceeds to detect dissimilar regions, given statistical decision criteria with respect to type I and type II errors [36].

III. REGISTRATION PROCEDURE

A. Data Space

In this paper, we consider pixel intensity values as our image features. This choice is appealing because it bypasses the segmentation of data into geometric primitives, a notoriously difficult problem for biomedical images. As mentioned in Section II-A, the use of pixel intensity values facilitates the inclusion of the entire informational content of the data. Moreover, the role of the reference and of the test volume can be exchanged at will. This allows one to check whether the inversion of the estimated “forward” registration is consistent with the estimated “backward” registration. This symmetry property is not necessarily taken for granted with some algorithms based on geometric primitives, for example, in [37]. Although raw intensity values are well suited to tasks like the detection of change or intramodality registration, their associated drawback is their lack of universality: They are not well suited to the problem of intermodality registration, a task in which one usually must resort to an intermediate feature representation, for example their gradient [38] or their histogram [33], [34], [39].

B. Criterion

Any automatic registration method requires the choice of an objective criterion that measures the similarity of the test data to the reference. As the optimization criterion, we select ε^2 , the integrated square difference of the intensity values, sometimes named the residue. Let f_R be the reference data and f_T the test data. Then, this criterion can be written as

$$\begin{aligned} \varepsilon^2 &= \int \int_{\{\mathbf{x}\} \subset R^q} (f_R(\mathbf{x}) - Q_{\mathbf{p}}\{f_T(\mathbf{x})\})^2 d\mathbf{x} \\ &= \|f_R(\mathbf{x}) - Q_{\mathbf{p}}\{f_T(\mathbf{x})\}\|^2 \end{aligned} \quad (1)$$

where $Q_{\mathbf{p}}\{f\}$ is a transformation parametrized by \mathbf{p} , and where q is the space dimension. Such a criterion lends itself well to minimization with respect to \mathbf{p} , and is well understood. In particular, this Euclidean dissimilarity measure is known to be maximum likelihood if the noise is additive, white, and Gaussian. Its drawback is a lack of robustness in the presence of severe outliers (e.g., nonstationary noise), where its minimum may become less pronounced. In the worst case, outliers predominate and the parameter \mathbf{p} for which ε^2 reaches its minimum can be quite different from \mathbf{p}_0 , the minimum in the noiseless case.

In medical images, outliers are always present, for example in the case of two PET images of the same brain at different functional states (certain brain areas are expected to display different activity levels). However, we do not expect these outliers to be dominant. In fact, the need for registration arises because the differences between the two brain images are

TABLE I
COMPOSITION RULES FOR TRANSLATION, AFFINE
TRANSFORMATION, AND CONTRAST CHANGE

$Q_p\{Q_q\{f(\mathbf{x})\}\}$	Q_p	T_a	A_A	C_a
Q_a	\circ			
T_b		$T_{b+a}\{f(\mathbf{x})\}$	$T_{Ab}\{A_A\{f(\mathbf{x})\}\}$	$T_b\{C_a\{f(\mathbf{x})\}\}$
A_B		$A_B\{T_{B^{-1}a}\{f(\mathbf{x})\}\}$	$A_{AB}\{f(\mathbf{x})\}$	$A_B\{C_a\{f(\mathbf{x})\}\}$
C_b		$C_b\{T_a\{f(\mathbf{x})\}\}$	$C_b\{A_A\{f(\mathbf{x})\}\}$	$C_{b+a}\{f(\mathbf{x})\}$

likely to be extremely small and cannot be detected without careful alignment [40].

C. Affine Transformation

As a first transformation of interest $Q_P\{f\}$, we consider the general affine transformation parametrized by a 3×3 matrix \mathbf{A} , by a translation vector \mathbf{b} and by a gray-level scaling factor γ . We include no gray-scale shift because it is virtually always true in medical images that some level (typically the background with value zero) has a physical interpretation, and should not be changed. This global transformation covers any combination of contrast change, translation, rotation around any center, skewing, shearing, and scaling. For our purpose, it is most convenient to decompose the transformation by using several operators, namely, a translation operator T_b , an affine operator A_A , and a contrast operator C_γ

$$\begin{cases} T_b\{f(\mathbf{x})\} = f(\mathbf{x} + \mathbf{b}) \\ A_A\{f(\mathbf{x})\} = f(\mathbf{A}\mathbf{x}) \\ C_\gamma\{f(\mathbf{x})\} = e^\gamma f(\mathbf{x}) \end{cases} \quad (2)$$

These operators are subject to the composition rules given in Table I. Using (2), we can express our first transformation as

$$Q_{b,A,\gamma}\{f(\mathbf{x})\} = T_b\{A_A\{C_\gamma\{f(\mathbf{x})\}\}\} = e^\gamma f(\mathbf{A}\mathbf{x} + \mathbf{b}). \quad (3)$$

D. Homomorphic Transformation

As an alternative, we propose a second global transformation with explicit restriction to the homomorphic case. It can accommodate contrast change, translation, rotation, and isometric scaling, but no skew or shear. To describe rotation we select a set of three Euler angles $\{\varphi, \theta, \psi\}$ and we let κ be our isotropic scaling parameter. Let us define the rotations

around the coordinate axes by

$$\begin{cases} \mathbf{A}_x(\omega) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\omega) & -\sin(\omega) \\ 0 & \sin(\omega) & \cos(\omega) \end{pmatrix} \\ \mathbf{A}_y(\omega) = \begin{pmatrix} \cos(\omega) & 0 & \sin(\omega) \\ 0 & 1 & 0 \\ -\sin(\omega) & 0 & \cos(\omega) \end{pmatrix} \\ \mathbf{A}_z(\omega) = \begin{pmatrix} \cos(\omega) & -\sin(\omega) & 0 \\ \sin(\omega) & \cos(\omega) & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{cases} \quad (4)$$

One of many possible descriptions of a rotation matrix \mathbf{R} by Euler angles is (5), shown at the bottom of the page. Note that the condition known as *gimbals lock* [41], where the representation of the rotation in terms of Euler angles is not unique, occurs far from $\{\varphi, \theta, \psi\} = \{0, 0, 0\}$. Also, one has to take care of the order of operations in the inverse of the previous rotation matrix

$$\mathbf{R}^{-1}(\varphi, \theta, \psi) = \mathbf{A}_z(-\psi)\mathbf{A}_y(-\theta)\mathbf{A}_x(-\varphi) \quad (6)$$

with

$$\mathbf{I} = \mathbf{R}^{-1}(\varphi, \theta, \psi)\mathbf{R}(\varphi, \theta, \psi) = \mathbf{R}(\varphi, \theta, \psi)\mathbf{R}^{-1}(\varphi, \theta, \psi). \quad (7)$$

We define the rotation operator $R_{\varphi,\theta,\psi}$ by

$$R_{\varphi,\theta,\psi}\{f(\mathbf{x})\} = f(\mathbf{A}_x(\varphi)\mathbf{A}_y(\theta)\mathbf{A}_z(\psi)\mathbf{x}), \quad \varphi, \theta, \psi \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right). \quad (8)$$

We define the isotropic scaling operator by

$$S_\kappa\{f(\mathbf{x})\} = f(e^\kappa \mathbf{x}). \quad (9)$$

Considering translation, rotation, scaling, and contrast change, we extend the composition rules of Table I as in Table II, where the explicit composition of two spatial rotations $R_{\varphi',\theta',\psi'}$ and $R_{\varphi'',\theta'',\psi''}$ involves $R_{\varphi,\theta,\psi}$ such that we obtain (10), shown at the bottom of the next page. Finally, we can express our homomorphic transformation by

$$\begin{aligned} Q_{b,\kappa,\varphi,\theta,\psi,\gamma}\{f(\mathbf{x})\} &= T_b\{S_\kappa\{R_{\varphi,\theta,\psi}\{C_\gamma\{f(\mathbf{x})\}\}\}\} \\ &= e^\gamma f(e^\kappa \mathbf{R}(\varphi, \theta, \psi)\mathbf{x} + \mathbf{b}). \end{aligned} \quad (11)$$

It is important to note that all the proposed operators are distributive with respect to addition, a property that we shall use later in this paper, as follows:

$$Q_P\{f(\mathbf{x}) + g(\mathbf{x})\} = Q_P\{f(\mathbf{x})\} + Q_P\{g(\mathbf{x})\}. \quad (12)$$

$$\begin{aligned} \mathbf{R}(\varphi, \theta, \psi) &= \mathbf{A}_x(\varphi)\mathbf{A}_y(\theta)\mathbf{A}_z(\psi) \\ &= \begin{pmatrix} \cos(\theta)\cos(\psi) & -\cos(\theta)\sin(\psi) & \sin(\theta) \\ \cos(\varphi)\sin(\psi) + \sin(\varphi)\sin(\theta)\cos(\psi) & \cos(\varphi)\cos(\psi) - \sin(\varphi)\sin(\theta)\sin(\psi) & -\sin(\varphi)\cos(\theta) \\ \sin(\varphi)\sin(\psi) - \cos(\varphi)\sin(\theta)\cos(\psi) & \sin(\varphi)\cos(\psi) + \cos(\varphi)\sin(\theta)\sin(\psi) & \cos(\varphi)\cos(\theta) \end{pmatrix} \end{aligned} \quad (5)$$

TABLE II
COMPOSITION RULES FOR TRANSLATION, ROTATION, SCALING, AND CONTRAST CHANGE

$Q_p\{Q_q\{f(\mathbf{x})\}\}$	Q_p	T_a	S_{κ^*}	$R_{\phi^*,\theta^*,\psi^*}$	C_a
Q_q	\circ				
T_b			$T_{b\phi^*}\{S_{\kappa^*}\{f(\mathbf{x})\}\}$		
S_{κ^*}		$S_{\kappa^*}\{T_{a\phi^*}\{f(\mathbf{x})\}\}$	$S_{\kappa^*}\{f(\mathbf{x})\}$	$S_{\kappa^*}\{R_{\phi^*,\theta^*,\psi^*}\{f(\mathbf{x})\}\}$	$S_{\kappa^*}\{C_a\{f(\mathbf{x})\}\}$
$R_{\phi^*,\theta^*,\psi^*}$			$R_{\phi^*,\theta^*,\psi^*}\{S_{\kappa^*}\{f(\mathbf{x})\}\}$	$R_{\phi^*,\theta^*,\psi^*}\{f(\mathbf{x})\}$	
C_b			$C_b\{S_{\kappa^*}\{f(\mathbf{x})\}\}$		

IV. OPTIMIZATION

We propose to solve $\partial\varepsilon^2(\mathbf{p})/\partial\mathbf{p} = 0$ with a new optimization scheme that we name ML*. It is a modification of the traditional Marquardt–Levenberg, an iterative gradient-based algorithm for nonlinear least-squares optimization problems [4]. In this section, we first present ML in its original form, and then proceed to show how it can be accelerated.

A. Traditional Marquardt–Levenberg (ML)

Fig. 2 depicts the traditional ML operative outline. At each step, f_T has to undergo the transformation $Q_{\mathbf{p}}$ before it is compared to f_R . More specifically, updating the actual set of M parameters \mathbf{p}_t requires the computation of an additive update component $\delta\mathbf{p}_t$ satisfying

$$\mathbf{p}_{t+1} = \mathbf{p}_t + \delta\mathbf{p}_t \quad (13)$$

with

$$\sum_{l=1}^M \alpha_{kl} \delta p_l = \beta_k \quad (14)$$

where $[\alpha_{kl}]_{M \times M}$ is derived through the Hessian matrix from the curvature matrix and where $[\beta_k]_{M \times 1}$ is proportional to the gradient of the residue. Approximation by a finite sum of the criterion given in (1) leads to

$$\varepsilon^2 \cong \chi^2(\mathbf{p}) = \frac{1}{N} \sum_{i=1}^N (f_R(\mathbf{x}_i) - Q_{\mathbf{p}}\{f_T(\mathbf{x}_i)\})^2 \quad (15)$$

where N is the number of pixels in the domain of interest and \mathbf{x}_i the coordinate of each pixel. We have

$$\begin{aligned} \beta_k &= \frac{-1}{2} \frac{\partial \chi^2(\mathbf{p})}{\partial p_k} \\ &= \frac{1}{N} \sum_{i=1}^N (f_R(\mathbf{x}_i) - Q_{\mathbf{p}}\{f_T(\mathbf{x}_i)\}) \frac{\partial Q_{\mathbf{p}}\{f_T(\mathbf{x}_i)\}}{\partial p_k} \end{aligned} \quad (16)$$

and

$$\begin{aligned} a_{kl} &= \frac{1}{2} \frac{\partial^2 \chi^2(\mathbf{p})}{\partial p_k \partial p_l} = \frac{1}{N} \sum_{i=1}^N \left(\frac{\partial Q_{\mathbf{p}}\{f_T(\mathbf{x}_i)\}}{\partial p_k} \frac{\partial Q_{\mathbf{p}}\{f_T(\mathbf{x}_i)\}}{\partial p_l} \right. \\ &\quad \left. - (f_R(\mathbf{x}_i) - Q_{\mathbf{p}}\{f_T(\mathbf{x}_i)\}) \frac{\partial^2 Q_{\mathbf{p}}\{f_T(\mathbf{x}_i)\}}{\partial p_k \partial p_l} \right). \end{aligned} \quad (17)$$

For reasons given in [42], the second derivative terms in (17) are usually ignored. Let us define

$$b_{kl} = \frac{1}{N} \sum_{i=1}^N \frac{\partial Q_{\mathbf{p}}\{f_T(\mathbf{x}_i)\}}{\partial p_k} \frac{\partial Q_{\mathbf{p}}\{f_T(\mathbf{x}_i)\}}{\partial p_l}. \quad (18)$$

Using (18), we characterize (14) by

$$\begin{cases} \alpha_{kl} = b_{kl}(1 + \lambda) & k = l \\ \alpha_{kl} = b_{kl} & k \neq l \end{cases} \quad (19)$$

where the free parameter $\lambda \geq 0$ determines degree to which the update $\delta\mathbf{p}_t$ conforms to a Newton method or to a steepest gradient approach. The characteristic of ML is to adapt λ at each iteration in such a way that the more successful the previous updates $\delta\mathbf{p}_t$ have been, the more Newton-like the next update $\delta\mathbf{p}_{t+1}$ will be. Conversely, the less successful, the more gradientlike the next update will be.

$$\begin{cases} \alpha_1 = \cos(\varphi') \sin(\theta'') + \sin(\varphi') \cos(\theta'') \sin(\psi'') \\ \alpha_2 = \sin(\varphi') \sin(\theta'') - \cos(\varphi') \cos(\theta'') \sin(\psi'') \\ \alpha_3 = \cos(\varphi'') \cos(\psi'') - \sin(\varphi'') \sin(\theta'') \sin(\psi'') \\ \alpha_4 = \cos(\varphi'') \sin(\psi'') + \sin(\varphi'') \sin(\theta'') \cos(\psi'') \\ \sin(\theta) = \alpha_1 \cos(\theta') + \sin(\theta') \cos(\theta'') \cos(\psi'') \\ \sin(\psi) = \frac{\sin(\psi')}{\cos(\theta')} (\cos(\theta') \cos(\theta'') \cos(\psi'') - \alpha_1 \sin(\theta')) - \alpha_2 \frac{\cos(\psi')}{\cos(\theta')} \\ \sin(\varphi) = \frac{\cos(\theta')}{\cos(\theta)} (\cos(\varphi') \sin(\varphi'') \cos(\theta'') + \alpha_3 \sin(\varphi')) - \alpha_4 \frac{\sin(\theta')}{\cos(\theta)} \end{cases}.$$

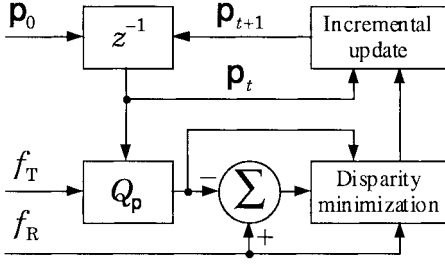


Fig. 2. Outline of the registration algorithm.

B. Modified Marquardt–Levenberg (ML*)

As both the vector $[\beta_k]$ and the matrix $[b_{kl}]$ (precursor to $[\alpha_{kl}]$) depend upon \mathbf{p} , and as \mathbf{p} changes from iteration to iteration, the standard implementation of ML (as presented above) requires that $[\beta_k]$ and $[\alpha_{kl}]$ be computed explicitly at each iteration. For our application, we show how to bypass these steps and save the associated computational cost. The general strategy is as follows. Instead of trying directly to find parameters \mathbf{p}_1 such that $\|f_R - Q_{\mathbf{p}_1}\{f_T\}\|^2 < \|f_R - Q_{\mathbf{p}_0}\{f_T\}\|^2$, we try to find \mathbf{p}_2 such that $\|Q_{\mathbf{p}_0}^{-1}\{f_R\} - Q_{\mathbf{p}_2}\{f_T\}\|^2 < \|Q_{\mathbf{p}_0}^{-1}\{f_R\} - f_T\|^2$. This new strategy is superior because the gradient of the criterion ε^2 with respect to \mathbf{p}_2 is now independent of the initial guess \mathbf{p}_0 and is computed about a fixed point in the parameter space. Using our definitions of the transformation operators and applying appropriate integration variable changes, it is easy to derive the following relations in 3-D:

$$\|f(\mathbf{x})\|^2 = \int \int_{\{\mathbf{x}\} \subset R^3} (f(\mathbf{x}))^2 d\mathbf{x},$$

$$\begin{cases} \|T_{\mathbf{b}}\{f(\mathbf{x})\}\|^2 = \|f(\mathbf{x})\|^2 \\ \|A_{\mathbf{A}}\{f(\mathbf{x})\}\|^2 = \frac{1}{|\mathbf{A}|} \|f(\mathbf{x})\|^2 \\ \|S_{\kappa}\{f(\mathbf{x})\}\|^2 = e^{-3\kappa} \|f(\mathbf{x})\|^2 \\ \|R_{\varphi, \theta, \psi}\{f(\mathbf{x})\}\|^2 = \|f(\mathbf{x})\|^2 \\ \|C_{\gamma}\{f(\mathbf{x})\}\|^2 = e^{2\gamma} \|f(\mathbf{x})\|^2. \end{cases} \quad (20)$$

Using the relations in (20) and the results of Tables I, II, and the distributive property (12), we can now write several equivalent forms of the criterion (1) for the affine case

$$\varepsilon^2 = \frac{e^{2\gamma}}{|\mathbf{A}|} \|C_{\Delta\gamma}\{A_{\mathbf{I}+\Delta\mathbf{A}}\{T_{\Delta\mathbf{b}}\{f_T(\mathbf{x})\}\}\} - C_{-\gamma}\{A_{\mathbf{A}^{-1}}\{T_{-\mathbf{b}}\{f_R(\mathbf{x})\}\}\}\|^2 \quad (21)$$

$$\varepsilon^2 = \frac{e^{2(\gamma+\Delta\gamma)}}{|\mathbf{A}(\mathbf{I}+\Delta\mathbf{A})|} \|f_T(\mathbf{x}) - C_{-\gamma-\Delta\gamma}\{A_{(\mathbf{I}+\Delta\mathbf{A})^{-1}\mathbf{A}^{-1}}\{T_{-\mathbf{b}-\mathbf{A}(\mathbf{I}+\Delta\mathbf{A})\Delta\mathbf{b}}\{f_R(\mathbf{x})\}\}\}\|^2 \quad (22)$$

$$\varepsilon^2 = \|T_{\mathbf{b}+\mathbf{A}(\mathbf{I}+\Delta\mathbf{A})\Delta\mathbf{b}}\{A_{\mathbf{A}(\mathbf{I}+\Delta\mathbf{A})}\{C_{\gamma+\Delta\gamma}\{f_T(\mathbf{x})\}\}\} - f_R(\mathbf{x})\|^2 \quad (23)$$

where the update in (13) is $\Delta\mathbf{p} = (\Delta\gamma, \Delta\mathbf{A}, \Delta\mathbf{b})^T$ for the

affine case. For the homomorphic case, we have

$$\varepsilon^2 = e^{2\gamma-3\kappa} \|C_{\Delta\gamma}\{R_{\Delta\varphi, \Delta\theta, \Delta\psi}\{S_{\Delta\kappa}\{T_{\Delta\mathbf{b}}\{f_T(\mathbf{x})\}\}\}\} - C_{-\gamma}\{R_{\varphi, \theta, \psi}^{-1}\{S_{-\kappa}\{T_{-\mathbf{b}}\{f_R(\mathbf{x})\}\}\}\}\|^2 \quad (24)$$

$$\varepsilon^2 = e^{2(\gamma+\Delta\gamma)-3(\kappa+\Delta\kappa)} \|f_T(\mathbf{x}) - C_{-\gamma-\Delta\gamma}\{(R_{\Delta\varphi, \Delta\theta, \Delta\psi}^{-1} \circ R_{\varphi, \theta, \psi}^{-1})\{S_{-\kappa-\Delta\kappa}\{T_{-\mathbf{b}-e^{\kappa+\Delta\kappa}\mathbf{R}(\varphi, \theta, \psi)\mathbf{R}(\Delta\varphi, \Delta\theta, \Delta\psi)\Delta\mathbf{b}}\{f_R(\mathbf{x})\}\}\}\}\|^2 \quad (25)$$

$$\varepsilon^2 = \|T_{\mathbf{b}+e^{\kappa+\Delta\kappa}\mathbf{R}(\varphi, \theta, \psi)\mathbf{R}(\Delta\varphi, \Delta\theta, \Delta\psi)\Delta\mathbf{b}}\{S_{\kappa+\Delta\kappa}\{(R_{\varphi, \theta, \psi} \circ R_{\Delta\varphi, \Delta\theta, \Delta\psi})\{C_{\gamma+\Delta\gamma}\{f_T(\mathbf{x})\}\}\} - f_R(\mathbf{x})\|^2. \quad (26)$$

Next, we show why these equivalent forms are useful. For example, in the affine case, finding a minimum to (21) with respect to $\Delta\mathbf{p} = (\Delta\gamma, \Delta\mathbf{A}, \Delta\mathbf{b})^T$ (incremental update) is equivalent to writing (23) with $\Delta\mathbf{p} = \mathbf{0}$ and then minimizing this equation with respect to \mathbf{p} . This yields $\delta\mathbf{p}$ according to (13) and corresponds to the standard ML. Minimizing (21) over $\Delta\mathbf{p}$ instead of the direct minimization of (23) over \mathbf{p} is advantageous because the curvature matrix $[b_{kl}]$ needs to be computed only once, at the parameter value $\Delta\mathbf{p} = \mathbf{0}$. The same is true for the derivatives $(\partial f_T / \partial \Delta\mathbf{p})|_{\Delta\mathbf{p}=\mathbf{0}}$. In short, ML solves $(\partial \varepsilon^2(\mathbf{q}) / \partial \mathbf{q})|_{\mathbf{q}=\mathbf{p}} = 0$ in an iterative fashion, while ML* solves $(\partial \varepsilon^2(\mathbf{p} + \mathbf{q}) / \partial \mathbf{q})|_{\mathbf{q}=\mathbf{0}} = 0$. In the former case, the point at which ε^2 is expanded into a Taylor series changes from one iteration to the next, while, in the latter case, this point is fixed.

After having adapted (14) for the minimization of (21) with respect to $\Delta\mathbf{p} = \mathbf{0} + \delta\mathbf{p}$ (see Appendix), one iteration of ML* runs as follows: Solve (14) and use (22) with $\mathbf{p} = \mathbf{p}_t$ and $\Delta\mathbf{p} = \delta\mathbf{p}_t$ to obtain a new estimate $\varepsilon^2(\mathbf{p}_{t+1})$. Update λ by the standard ML procedure. Compute \mathbf{p}_{t+1} using the updating rules found in (23), test for convergence and go to the next iteration if necessary. Otherwise, the final transformation is given by (23).

Applying these operations to the homomorphic case is straightforward: Replace (21), (22) and (23), respectively, by (24), (25), and (26). More precisely, write the set of equations minimizing (24) and solve them by using (14), which yields a potential update given by $\delta\mathbf{p} = (\Delta\mathbf{b}, \Delta\kappa, \Delta\varphi, \Delta\theta, \Delta\psi, \Delta\gamma)^T$. The next step is to transform the reference image f_R according to (25). This equation describes how to combine the old parameters \mathbf{p} and the update $\delta\mathbf{p}$ when applied to f_R (e.g., the rotation would be given by the composition $(R_{\Delta\varphi, \Delta\theta, \Delta\psi}^{-1} \circ R_{\varphi, \theta, \psi}^{-1})$, the scaling would be given by $(-\kappa - \Delta\kappa)$, etc.). Then, evaluate the residue between the test image f_T and the inverse transformed reference image $Q_{\mathbf{p}^{-1}}\{f_R\}$. This residue needs to be scaled by $e^{2(\gamma+\Delta\gamma)-3(\kappa+\Delta\kappa)}$ to yield the correct value ε^2 for the criterion (1), or, equivalently, (26). Based on the comparison between this new ε^2 and the best one observed so far, and applying the considerations of Section IV-C, decide whether the potential update should be rejected or not. If the iteration is successful, then update \mathbf{p} according to (26). It is this equation

which describes how to combine the old parameters \mathbf{p} and the update $\delta\mathbf{p}$ when applied to f_T (e.g., the rotation would be now given by the composition $(R_{\varphi,\theta,\psi} \circ R_{\Delta\varphi,\Delta\theta,\Delta\psi})$, the scaling would be given by $(\kappa + \Delta\kappa)$, etc.). When convergence is reached, transform the test image f_T according to (26) with the best set of parameters observed so far.

In summary, we update the inverse transformation $Q_{\mathbf{p}-1}$ that is applied to f_R instead of the direct transformation $Q_{\mathbf{p}}$ that is applied to f_T . Note that the optimization process requires explicit knowledge of the partial derivatives $\partial f_T(Q_{\mathbf{p}}(\mathbf{x}))/\partial\mathbf{p} = (\partial f_T(Q_{\mathbf{p}}(\mathbf{x}))/\partial\mathbf{x})(\partial Q_{\mathbf{p}}(\mathbf{x})/\partial\mathbf{p})$, where $f_T(Q_{\mathbf{p}}(\mathbf{x}))$ is an equivalent notation for $Q_{\mathbf{p}}\{f_T(\mathbf{x})\}$. Given our transformation (either affine or homomorphic), we determine analytically the dependence $\partial Q_{\mathbf{p}}(\mathbf{x})/\partial\mathbf{p}$ from (21) or (24), respectively (see Appendix), while we compute the spatial gradient $\partial f_T(Q_{\mathbf{p}}(\mathbf{x}))/\partial\mathbf{x}$ from the coefficients of the exact cubic spline fit to the data using the digital filtering techniques described in [43], [44].

C. Convergence

We consider three concurrent criteria for deciding when to stop the iteration process. The first is self-evident: We stop when a near-perfect match is met, that is, when $\chi^2 \leq T_1$, where T_1 is some *a priori* threshold. The second uses the observed relative gain $\Delta\chi^2/\chi^2$ at each successful iteration step: We declare convergence to be reached whenever this gain is below another *a priori* threshold T_2 . The third one uses $|\delta p_k/p_k|$, the relative change of parameter values at each iteration step: When at least one of the parameters is still changing, we continue; but when all are no longer changing, we stop. If any parameter p_k happens to be zero, or near zero, we substitute $|\delta p_k/p_k| \leq T_3$ by a similar, although possibly less effective criterion extracted from some of the parameters of the algorithm. Specifically, we use $|b_{kk}\delta p_k/\beta_k| \leq T'_3$, which behaves asymptotically like $1/\lambda$ for increasing λ . If, by mishap, or because an optimum has been nearly reached, even the gradient β_k is too small for this last criterion to be applicable, we iterate anyway and rely on the two previous criteria to eventually determine that convergence has been really reached.

D. Practical Issues

We use the technique of singular value decomposition for the solution of the system of linear equations in (14). Apart from robustness, its advantage is that it allows for a reduction in rank of the matrix $[\alpha_{kl}]$. Such reduction may arise, for example, when a parameter is forced to an arbitrarily fixed value. This last feature allows to tune the whole algorithm by independently switching on or off the optimization of any of the parameters. Typically, partial optimization is useful when dealing with images instead of volumes, because the parameters needed for the description of a two-dimensional (2-D) transformation are a subset of those needed for 3-D. For example, the rotation of an image is equivalent to the rotation of a volume with $\varphi = 0$, $\theta = 0$, with ψ as the only free rotation parameter.

There are two points to be addressed regarding our formulation of ML*. The first concerns the substitution of criterion (1) by its discrete version (15). At first sight this seems unnecessary because we have at our disposal a continuous model of the data (in our case, cubic spline polynomials), but it is nevertheless beneficial both in a practical and a theoretical sense. The practical benefit is an obvious computational simplification: The explicit integration of splines requires more operations than does a simple addition of terms. In any case, the discrete version is very close to the continuous one [6]. The theoretical benefit arises from the fact that the data set is finite: In (1) we perform integration over the entire domain of real numbers (L_2 norm). In practice, however, the domain is always limited by the size of the image. By introducing a finite domain of integration, the Leibnitz rules for derivatives are required in the computation of $\partial\epsilon^2(\mathbf{p})/\partial\mathbf{p}$, which then becomes explicitly dependent on the integrand in (1), evaluated at the borders. This last requirement is undesirable; intuitively, one would prefer to see all the contributions in (1) to be equally weighted, which is exactly what the discrete version (15) does. In fact, the discrete implementation allows for a controlled weighting of the measurements. For example, the original ML formulation uses weights inversely proportional to the measurement variance. In our application, we use binary masking whenever convenient, which corresponds to unitary or infinite variance. The mask is transformed in the same way as the data (albeit with nearest-neighbor interpolation).

The second point concerns the validity of the relations in (20) (involving for example $A_{\mathbf{A}}$ and S_{κ}), which hold formally only if one considers an infinite domain of integration. We argue that ML* holds even under finite data (or masking), provided that at least one of the two following conditions is satisfied: either the scaling factor κ or the determinant $|\mathbf{A}|$ is close enough to unity, or sufficient data are available. In the latter case, convergence will occur for the same values in \mathbf{p} , whether the criterion ϵ^2 is expressed in the regular (ML) or in the modified (ML*) form. In most applications, both conditions are usually satisfied.

V. INTERPOLATION MODEL

Interpolation is a model-based process that allows one to recover the value of a function from a given set of its samples, within a given abscissa range. Here, we have chosen to make use of a separable model because such a model is easy to manipulate. It is given by

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in Z^q} c(\mathbf{k}) \prod_{i=1}^q \varphi(x_i - k_i) \quad (27)$$

where $c(\mathbf{k})$ is a discrete set of coefficients and $\varphi(x)$ is a continuous synthesis function. For example, [45] has proposed to use $\varphi(x) = \text{sinc}(x)$. This choice limits $f(\mathbf{x})$ to being a finite-bandwidth function with coefficients given by $c(\mathbf{k}) = f(\mathbf{k})$. We prefer to use $\varphi(x) = \beta^{(n)}(x)$, where $\beta^{(n)}$ is a symmetrical spline of degree n . In this case, $f(\mathbf{x})$ is a piecewise polynomial function, and the model is especially easy to manipulate both analytically and numerically [43]. In particular, the set of coefficients $c(\mathbf{k})$ can be computed

efficiently by separable recursive linear filtering. This spline model, although it uses a synthesis function of finite support, is equivalent to a model that uses a cardinal spline of the same degree n , which has an infinite support for $n \geq 2$. As the degree of the cardinal spline tends to infinity, Shannon interpolation and spline interpolation become equivalent [7].

In this paper, we select $n = 3$ because it has several advantages. First, a cubic cardinal spline is already quite close to a sinc function, at a computational cost that is lower than would be required within the traditional approach of truncation and apodization. In this view, it is well suited to processing data that are approximately bandlimited by nature or by construction. Second, even though a cubic cardinal spline is of infinite extent, it does not introduce excessive ringing, which makes it an attractive candidate in this type of applications. Third, it offers the lowest possible cost at which we can implement ML^* in a consistent fashion. This follows because we need to compute the gradient of the image with respect to the spatial coordinates sampled at the grid points in order to compute the gradient of ε^2 with respect to the registration parameters. Recall that, for splines of degree zero this gradient is undefined at the sampling points, while it is ill-defined for splines of degree one (the data would have to be sampled at the discontinuities). The use of quadratic splines would suffice, but it introduces a computational cost exactly as high as in the case of our choice, degree three, or $\beta^{(3)}$, which has been shown to be a good compromise between quality and speed for a variety of other tasks as well.

VI. PYRAMIDS

A. Multiresolution Processing

We take advantage of the iterative nature of the algorithm by introducing a change of resolution between one or another of the iterations. For practical purposes, the quasi-Newtonian optimization method that we use can converge in very few steps (ideally, one step) if the previous estimate is sufficiently good. This is often the case for all but the coarsest resolution level. Consequently, most of the iterations are required at the coarsest level where, due to the data decimation inherent in the pyramid, the number of pixels is small. This results in a considerable savings of computation time.

A second very important advantage of using a resolution pyramid is that the smoothness conditions imposed by the polynomial splines tend to regularize the optimization problem by causing the surface $\varepsilon^2(\mathbf{p})$ to become smoother at coarser resolutions. The antialiasing filters involved in the downsampling effectively remove more and more of the image detail and noise. As a result, the algorithm achieves first a registration with respect to the large-scale features in the data, and then only makes small corrections for progressively finer details. For most images, it is very likely that the approximate location of the global optimum will not be missed by the solution at the coarsest level, whereas an attempt to find the solution by considering the finest level only could more easily become trapped into a false local minimum. This is important because

we make no use of stochastic procedures that would facilitate the escape from local minima.

We construct a dyadic pyramid by starting at some actual (fine) level, of which we compute a coarser representation for the next level. We take advantage of the spline model to minimize in a least-squares sense the difference between the fine and the coarse representations, and we then proceed recursively. This implementation of the pyramid construction is based on [46], using cubic splines.

B. Consistency

We feel that it is important to enforce the processing of data in a way that is internally consistent. In particular, there are four main tasks where this applies. The first one is the computation of a pyramid. Here, we apply the technique presented in [46], which provides a sequence of fine-to-coarse approximations that are optimal in a least-squares sense. The second task is the computation of derivatives; as discussed in Section V, we have selected a model with an order high enough to enable exact computation. Note that, in the case of the standard ML , the requirements for the spline order are less severe: There, we perform the derivative computations at many points in the parameter space, thus sampling the gradient values at points in data space that do not lie exactly on the grid. It follows that inaccuracies tend to even out in the course of optimization, and the problem of ill-defined gradient arises infrequently. By contrast, in the ML^* case, we perform computations once only, at the fixed-point of identity in the parameter space, which requires to sample the gradient exclusively on the grid. This allows the consideration of a spline of degree one as a possible model for ML , while barring it for ML^* . The third task is the application of geometric transformations during the course of optimization; there, we use a resampling scheme in order to apply the discrete criterion (15) exactly, even though some geometric scaling might be present in the transformation. Finally, the last task is, strictly speaking, not within the scope of registration, for it consists in the production of a transformed test image once the transformation parameters have been determined. However, in practice, this operation is often desired, and we include it for completeness. For this last transformation, we again select the model used during optimization, and again resample because we want to produce the very data that minimizes the selected criterion. Supposing the registration has been successful, any additional process (e.g., change of model order, antialiasing filtering, etc.), would necessarily result in an output that no longer minimizes the criterion for which the registration parameters have been determined.

C. Propagation of Parameters Between Levels

Since our method is implemented by a multiresolution approach, the derivatives and the curvature matrices have to be computed at each resolution level, and the transformation parameters need to be propagated between levels. To illustrate this propagation, consider an initial geometric transformation

$T_b\{A_A\{f(\mathbf{x})\}\}$ given by

$$\begin{cases} y_1 = b_1 + a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \\ y_2 = b_2 + a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \\ y_3 = b_3 + a_{31}x_1 + a_{32}x_2 + a_{33}x_3. \end{cases} \quad (28)$$

Consider now the scale factors (s_1, s_2, s_3) that correspond to the conversion from one resolution level to the next; typically, we have $s_i \in \{\frac{1}{2}, 1, 2\}$. Applying these scale factors, one obtains a new, equivalent transformation $T_{b'}\{A_{A'}\{f(\mathbf{x}')\}\}$ as in

$$\begin{cases} y'_1 = b'_1 + a'_{11}x'_1 + a'_{12}x'_2 + a'_{13}x'_3 \\ y'_2 = b'_2 + a'_{21}x'_1 + a'_{22}x'_2 + a'_{23}x'_3 \\ y'_3 = b'_3 + a'_{31}x'_1 + a'_{32}x'_2 + a'_{33}x'_3 \\ \begin{cases} x'_1 = s_1x_1 \\ x'_2 = s_2x_2 \\ x'_3 = s_3x_3 \end{cases}, \quad \begin{cases} y'_1 = s_1y_1 \\ y'_2 = s_2y_2 \\ y'_3 = s_3y_3 \end{cases} \end{cases} \quad (29)$$

These four systems of three equations are satisfied by

$$\begin{cases} b'_1 = s_1b_1 \\ b'_2 = s_2b_2 \\ b'_3 = s_3b_3 \end{cases}, \quad \begin{cases} a'_{11} = a_{11} & a'_{12} = \frac{s_1}{s_2}a_{12} & a'_{13} = \frac{s_1}{s_3}a_{13} \\ a'_{21} = \frac{s_2}{s_1}a_{21} & a'_{22} = a_{22} & a'_{23} = \frac{s_2}{s_3}a_{23} \\ a'_{31} = \frac{s_3}{s_1}a_{31} & a'_{32} = \frac{s_3}{s_2}a_{32} & a'_{33} = a_{33} \end{cases} \quad (30)$$

The last equations show how a description of the geometric transformation is carried from one level of the pyramid to the next. In general, the relations in Table I or Table II are needed to determine the additional translation correction that may arise when the coordinate system is displaced during a change of level.

VII. EXPERIMENTS

A. Ideal Case

We begin our series of experiments with an ideal case: The test and reference sets are identical except for an affine transformation, and no noise is present. We consider the 256×256 image of Fig. 3 instead of a volume. Our goal here is to compare ML to ML*, both in terms of precision and speed. For the comparison to be fair, we have modified the convergence criterion presented at Section IV-C so that a preset number of iterations n_i are executed at each level i in the pyramid, specifying the same number for ML and ML*. Let $i = 0$ designate the finest level in the pyramid, and set $n_{i+1} = 2n_i$, $i \geq 0$. This strategy ensures that a sufficient number of iterations are performed at the coarsest level, as the initial conditions may be far from optimal at this particular level. At finer levels, the number of iterations can be sharply reduced because only a few iterations are necessary when the initial conditions have been properly estimated. Let t_0 be the time spent on a single iteration at the finest level. Ignoring the computational overheads (e.g., establishing the resolution pyramid, or computing the β -spline coefficients, or computing the final transformation), we predict a total execution time t to be given by

$$t = \sum_{i=0}^{N-1} n_0 t_0 2^{(1-q)i} = \frac{n_0 t_0 (1 - 2^{(1-q)N})}{1 - 2^{(1-q)}} \quad (31)$$

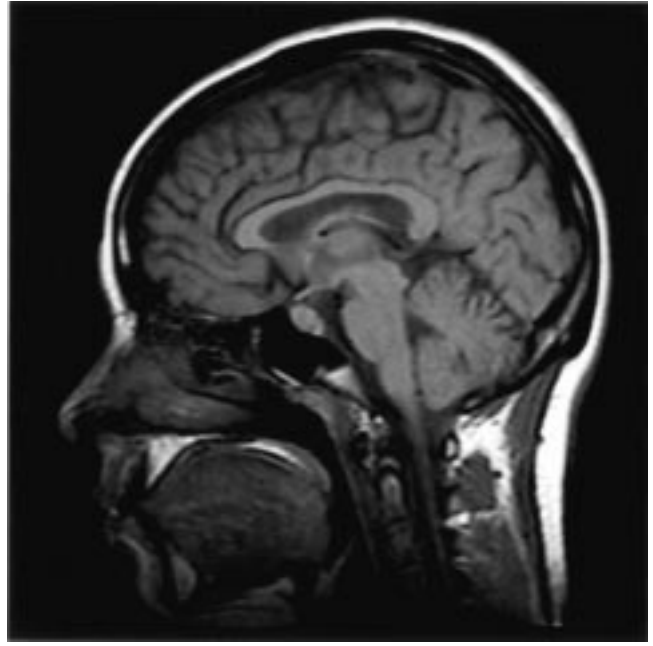


Fig. 3. MRI slice.

where N is the number of levels in the pyramid and q the space dimension. In particular, for an infinite pyramid and for 2-D images we have $t = 2n_0 t_0$. For 3-D volumes we have $t = \frac{4}{3}n_0 t_0$.

In order to test this prediction, we first generate a series of random transformations, each consisting of a rotation with a uniformly distributed angle $\theta' \in [-\frac{\pi}{36}, \frac{\pi}{36}]$, and with an independent and uniformly distributed translation of each coordinate $\Delta' \in [-2.5, 2.5]$. In order to remove experimental bias, we apply this transformation twice to the image of Fig. 3; a first application of the transformation inverse produces the test image f_T , and a second application of the direct transformation produces the reference image f_R . With this methodology, neither ML nor ML* algorithm is favored. Also, the transformation to recover, being the square of the initial random transformation, has twice the range of the original random transformation: $\theta \in [-\frac{\pi}{18}, \frac{\pi}{18}]$ and $\Delta \in [-5, 5]$. In addition, we apply these transformations by resampling a high-order spline model ($\beta^{(7)}$). This is essentially equivalent to sinc interpolation. We then attempt to register the test image f_T to its reference f_R , which produces an estimated set of parameters $\tilde{\mathbf{p}}$. Note that we do not try to recover the rotation exactly; rather, we search for a general affine transformation. We generated 100 random transformations per experiment and pooled the results.

Fig. 4 shows the behavior of the ML* algorithm when we use five levels and two iterations at the finest level. The two curves show the time spent at each level, in seconds on a Sparc20, and the residue between the transformed test image and the reference after convergence within a given level. This last value is computed on the full-scale version of the images. We see that, although the number of iterations is halved by going from one level of the pyramid to the next, the time spent at each level nevertheless increases sharply. We also see that the minimization algorithm performs well, reducing the initial

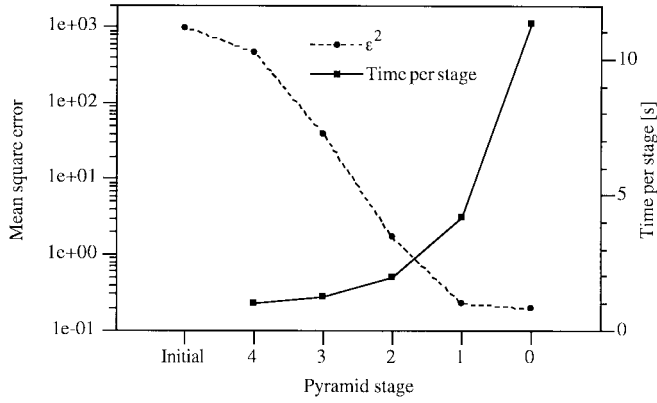


Fig. 4. Pooled performance of 100 runs of ML* with $N = 5$ and $n_0 = 2$.

TABLE III
MULTIRESOLUTION BEHAVIOR OF THE ALGORITHMS

Stage	Iterations	ML1	ML3	ML*3	ML1	ML3	ML*3
		ϖ [pxl]	ϖ [pxl]	ϖ [pxl]	t_i [s]	t_i [s]	t_i [s]
initial		4.5440	4.5440	4.5440			
4	32	3.2262	1.6922	1.6424	00.70	01.36	01.00
3	16	1.3303	0.3911	0.4809	00.95	01.92	01.23
2	8	0.3059	0.1099	0.0932	01.67	03.32	01.98
1	4	0.0454	0.0109	0.0109	03.43	06.67	04.18
final	2	0.0049	0.0010	0.0009	08.13	15.43	11.32
					14.88	28.70	19.71

mean square error by more than three orders of magnitude along the way.

Our next step is to examine the registration performance from a geometric point of view. To this end, we introduce a warping index ϖ (an average geometric error), which we compute by comparing the true transformation to its estimate. In general, the true transformation is unknown (if it were known, there would be no need for registration). However, in our experiment design, we can compare the estimated transformation $\tilde{\mathbf{p}}$ and the true transformation \mathbf{p} on the basis of

$$\varpi = \frac{1}{\text{Card}(V)} \sum_{\mathbf{x} \in V} \|\mathbf{p}(\mathbf{x}) - \tilde{\mathbf{p}}(\mathbf{x})\|. \quad (32)$$

Table III presents the pooled results of 100 registration experiments per algorithm, using the same set of random transformations for each. The three algorithms that we consider are ML1 (standard Marquardt–Levenberg, bilinear interpolation), ML3 (standard Marquardt–Levenberg, bicubic interpolation), and ML*3 (modified Marquardt–Levenberg, bicubic interpolation). Here, by standard ML, we mean Marquardt–Levenberg in a multiresolution framework.

In terms of registration accuracy, we conclude from Table III that ML3 and ML*3 are essentially equivalent, while the accuracy obtained by ML1 is about fivefold lower. Two mechanisms explain this difference. First, within a given level of the pyramid, the accuracy is principally dependent on the resampling precision or, equivalently, the model order. It is also influenced by the definition of the derivatives used for determining the gradient vector and the Hessian matrix. This last issue is less important in the case of ML because the derivatives are computed as many times as there are iterations.

TABLE IV
DEPENDENCE ON THE NUMBER OF ITERATIONS

Strategy	n_0	ML1	ML3	ML*3	ML1	ML3	ML*3
		ϖ [% pxl]	ϖ [% pxl]	ϖ [% pxl]	t [s]	t [s]	t [s]
16.8.4.2.1	1	4.28	0.70	0.76	09.6	18.8	15.2
32.16.8.4.2	2	0.49	0.10	0.09	14.8	28.7	19.7
64.32.16.8.4	4	0.48	0.10	0.08	25.3	48.8	29.0
128.64.32.16.8	8	0.44	0.12	0.08	46.3	88.4	47.0

This tends to average out their inaccuracies. However, in the case of ML*, an accurate definition of the derivatives is crucial because they are computed only once. The second mechanism is important when switching from one level of the pyramid to the next. There, a high-order model ensures better consistency in locating the optima across levels because less aliasing is present in the coarser pyramid levels as compared to a lower order model. This results in a better propagation of the initial conditions, which means that the superlinear convergence mode of the Marquardt–Levenberg algorithm is reached earlier with high-order models than with low-order ones.

In terms of speed, the more iterations, the more we expect ML* to be faster than ML. Comparing similar models (ML3 and ML*3), we see that this advantage shows up clearly even for as few as two iterations. Although we had left out any impact of the computational overheads in our prediction (31), we see that the relation $t = 2n_0 t_0$ is in rough agreement with the experimental data. Clearly, a substantial amount of time is devoted to iterations at the finest scale. For this reason, it is very important to have excellent starting conditions for this last level of the pyramid, so that a small number of iterations is sufficient to reach convergence. If these good starting conditions are indeed available, then the super linear property of ML or ML* will assure even further benefits. For this reason, we feel it is especially important to make use of high-order models right from the beginning. If time is critical, one should at least use such models for the initial levels of the pyramid, where they come cheap, and perhaps switch to a lower model for the final level (this approach is not pursued here).

Table IV presents the results of varying the number of iterations (only the results after the final level are given). Here again, each entry results from the pooling of 100 experiments. In this table, we see that, for $n_0 \geq 2$, the further iterations bring no marked increase in registration accuracy, while it incurs an appreciable lengthening of computation time. We explain this behavior by the fact that neither ML nor ML* is a stochastic algorithm; there is no possible escape from a local minimum. Once such a local minimum has been found, these algorithm will stay there; no matter how many additional iterations are performed, the accuracy will not increase.

In summary, we conclude from these experiments that the most economical way to achieve high registration accuracy is to increase the interpolation order, rather than to increase the number of iterations.

B. Noisy Case

Here, we perform essentially the same experiments as in the previous section, but we add independent realizations of white

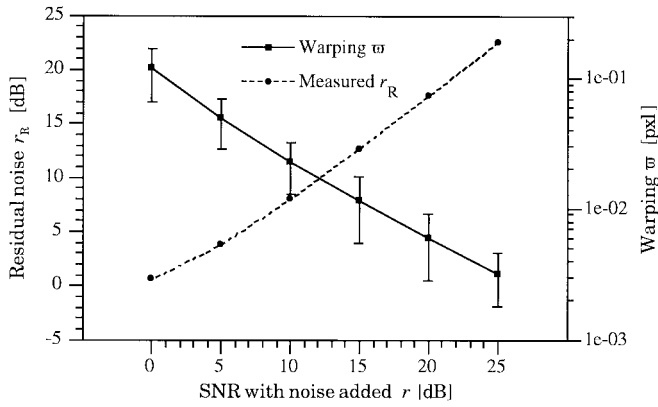


Fig. 5. Pooled performance of 100 runs of ML* with noise added.

Gaussian noise to both the reference and test images before their registration. We report the amount of added noise as a signal-to-noise ratio (SNR) r expressed in dB, according to the following definition:

$$r = 10 \log_{10}(\sigma^2(f_R)/\sigma^2(f_N)) \quad (33)$$

where $\sigma^2(f_R)$ is the squared variance of the original signal of Fig. 3, and where $\sigma^2(f_N)$ is the squared variance of the added noise. After registration, we compute the SNR between the transformed test data $Q\{f_T\}$ and the reference f_R by

$$r_Q = 10 \log_{10}(\sigma^2(f_R + f_{N_R}) / \sigma^2(f_R + f_{N_R} - Q\{f_T + f_{N_T}\})). \quad (34)$$

Fig. 5 presents the results of this experiment. We emphasize that, different from [5], we do not satisfy ourselves with estimates of the statistical uncertainties in the accuracy of the match. Because we use the identical starting set of image data for both test and reference, we know the true error, as measured though the warping index w introduced at (32). Fig. 5 shows that its growth is close to being proportional to the SNR r given in (33). In this same figure, we have reported the standard deviation of w across runs; this last value is a valid indicator of the dispersion of the measurements. We experienced no outliers, even when the variance of the noise was as large as the signal variance (0 dB case). Fig. 6 shows the original image of Fig. 3, corrupted by exactly this amount of noise. In this case, it is worth noting that, on average, ML* still achieves registration within about a tenth of a pixel. This accuracy comes at a cost, however: In the case of the noisy experiments of this section, we have used the convergence criterion described in Section IV-C. Altogether, for $r \geq 10$ dB, we find a convergence time comparable to the case $n_0 = 2$ in Table IV. For noisier images however, the time needed to reach convergence increases significantly; at $r = 0$ dB, it is equivalent to the noiseless case $n_0 = 8$.

C. Intersubject 3-D PET Brain Scans

In the previous experiments, we showed cases where the deformation model (affine transformation) was consistent with

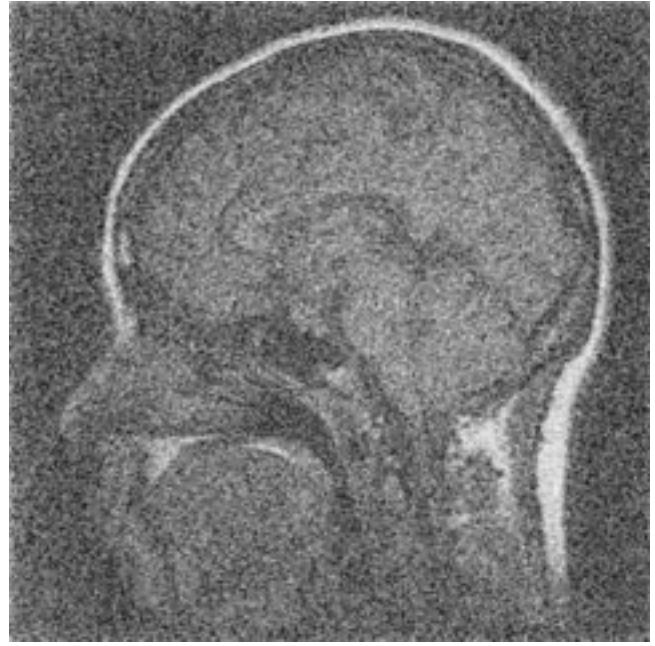


Fig. 6. MRI slice corrupted with white Gaussian noise at a 0 dB level.

the data. In the following experiment, this assumption is no longer valid. We consider brain volumes from different subjects acquired by a PET imaging device. These brains exhibit not only size differences but also shape differences. The goal now is to register these different brains to a common reference (a subject selected *a priori*), notwithstanding the fact that the dissimilarities in shape may introduce outliers in the statistics of the difference between the reference and the test volumes. Fig. 7 shows five unregistered slices from several subjects, cut at the same nominal position in the original volumes. At the top left is the reference slice, and at the bottom right is the average of the 32 selected slices for a set of subjects. Fig. 8 shows the same configuration after 3-D registration to the reference volume has been performed before averaging.

PET images are typically noisy, so we need to mask out most of the background in the image, which explains the disappearance of streaks when comparing Fig. 7 to Fig. 8. As the detailed outline of the mask influences the registration accuracy in a negligible way, the masking procedure can be extremely crude (it does not matter if some background is present or if some small amounts of brain are disregarded). As the masks can be specified independent of the data, any masking procedure can be used. Here, we applied a very lowpass Gaussian filter to the image, performed binary Max-Lloyd quantization, and then computed a pyramid in the same way as for the data, masking out the values below the Max-Lloyd threshold at each level. Regarding the success of the registration, it is evident that most of the brain features of interest are far better resolved in the registered than in the unregistered average volume, which are given at the lower right of Figs. 8 and 7, respectively. This indicates that our least-squares registration criterion is robust enough to cope with deformations that cannot be modeled by an affine transformation. We note that these deformations are,

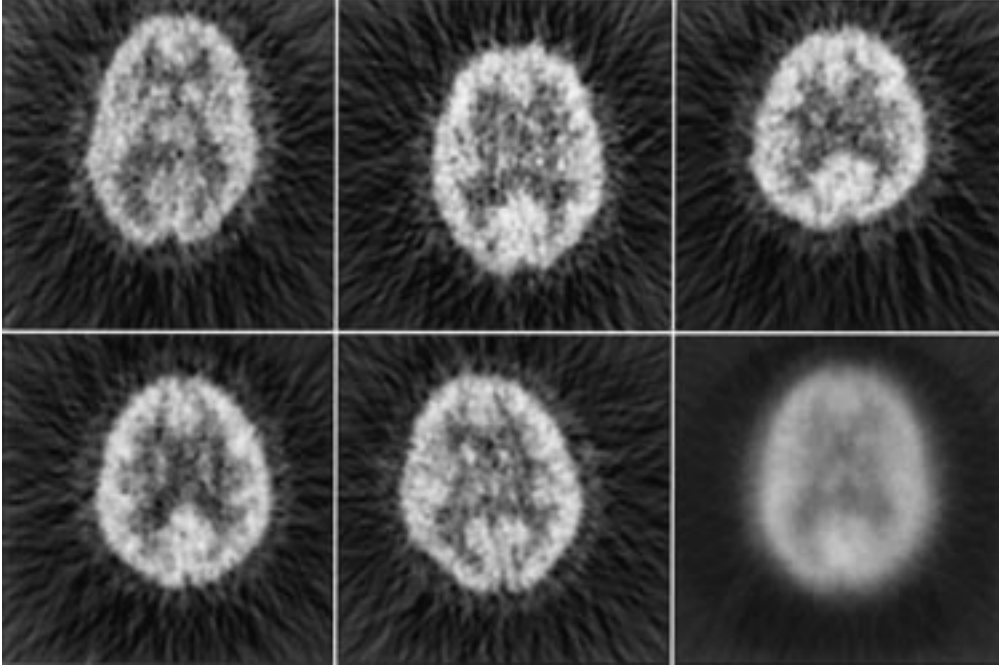


Fig. 7. Unregistered slices and their average.

in fact, hard to capture globally; even when using a nonlinear model with 105 parameters, nontrivial anatomical intersubject differences may persist [47]. Regarding speed, a $128 \times 128 \times 21$ pair of volumes is completely registered in about 3–4 min on a SPARC20, including 10 s spent in computing the final transformation (that ignores interpolation outside the mask), and less than half a minute spent in computing the two pairs of volume pyramids (one pair for the data and one pair for the masks). Without masking, the computation time of a transformation at the finest level of the pyramid is half a minute.

VIII. CONCLUSION

We have described a fully automatic registration algorithm that uses the original gray levels as elements in the feature space, and considers a Euclidean least-squares criterion for the simultaneous determination of a general 3-D affine transformation and a linear change of contrast. We have introduced a new search strategy that takes advantage of a resolution pyramid and implements a variation of the Marquardt–Levenberg algorithm for nonlinear least-squares optimization. Great care has been taken to ensure the consistency of all steps of the procedure by using the same cubic spline representations throughout. We have implemented this algorithm, presented several experiments involving 2-D images and 3-D volumes, and were able to show good performance with respect to accuracy and speed.

An attractive feature of this registration algorithm is that it can easily incorporate *a priori* knowledge by way of a (possibly multivalued) mask. Another attractive feature is that the general affine transformation can be constrained to be a rigid-body transformation, with or without scaling, or even to a simple translation. Further, the method, developed for a 3-D

case, can be applied without any modification to a 2-D case as well. Finally, no restricting assumptions regarding the data are made, and no landmark estimation is required. This results in an all-purpose, robust registration method. Without requiring any parameter tuning, the same algorithm has been applied to PET data and to MRI and fMRI data, and to many other data sets not reported here.

APPENDIX

In this appendix, we derive the modified Marquardt–Levenberg algorithm for the affine transformation introduced in Section III-C (the treatment of the homomorphic transformation case is similar, but would result in lengthier expressions). We consider expression (21) and discretize according to (15)

$$\chi^2(\mathbf{p}) = \frac{c^{2\gamma}}{|\mathbf{A}|} \sum_{i=1}^N (C_{\Delta\gamma} \{A_{\mathbf{I}+\Delta\mathbf{A}} \{T_{\Delta\mathbf{b}} \{f_T(\mathbf{x}_i)\}\}\} - C_{-\gamma} \{A_{\mathbf{A}^{-1}} \{T_{-\mathbf{b}} \{f_R(\mathbf{x}_i)\}\}\})^2 \quad (\text{A.1})$$

where the summation is over all pixel values. We compute the gradient $\partial\chi^2(\mathbf{p})/\partial\Delta\mathbf{p}$, remembering from the discussion in Section IV-C that $\Delta\mathbf{p} = \mathbf{0}$, and introduce it into expression (16)

$$\beta_k = \frac{-1}{2} \frac{\partial\chi^2(\mathbf{p})}{\partial\Delta\mathbf{p}_k} = \frac{-c^{2\gamma}}{|\mathbf{A}|} \sum_{i=1}^N (f_T(\mathbf{x}_i) - C_{-\gamma} \{A_{\mathbf{A}^{-1}} \{T_{-\mathbf{b}} \{f_R(\mathbf{x}_i)\}\}\}) \frac{\partial Q_{\Delta\mathbf{p}} \{f_T(\mathbf{x}_i)\}}{\partial\Delta\mathbf{p}_k}. \quad (\text{A.2})$$

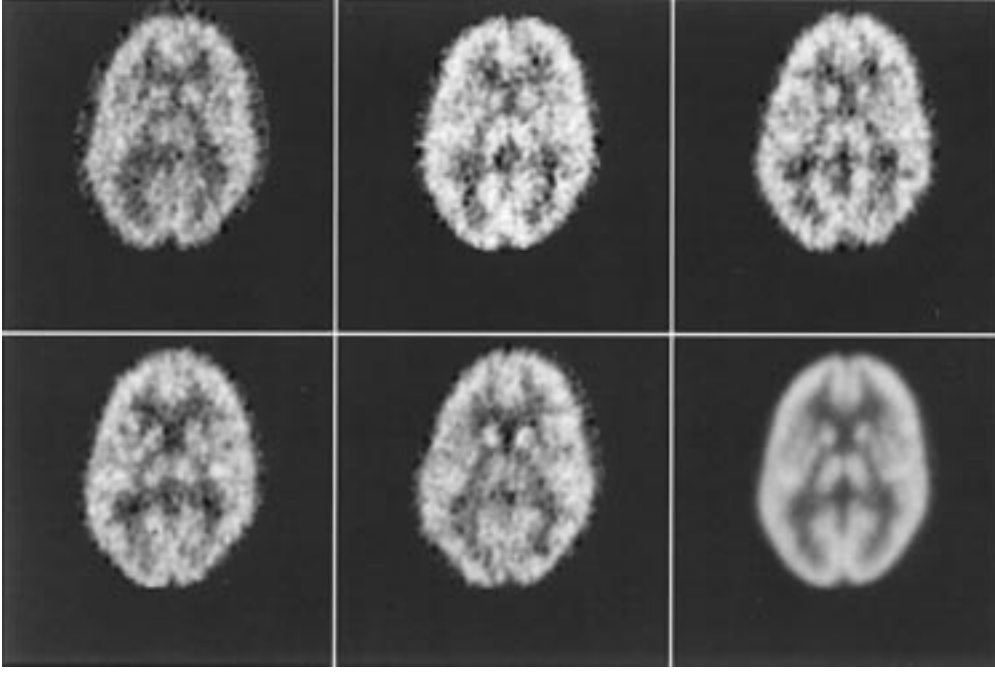


Fig. 8. Registered slices and their average.

Explicitly, we have

$$\begin{aligned}
 Q_{\Delta \mathbf{p}}\{f_T(\mathbf{x})\} &= e^{\Delta \gamma} f_T \left(\begin{pmatrix} 1 + \Delta a_{11} & \Delta a_{12} & \Delta a_{13} \\ \Delta a_{21} & 1 + \Delta a_{22} & \Delta a_{23} \\ \Delta a_{31} & \Delta a_{32} & 1 + \Delta a_{33} \end{pmatrix} \right. \\
 &\quad \times \left. \begin{pmatrix} x + \Delta x \\ y + \Delta y \\ z + \Delta z \end{pmatrix} \right) \quad (A.3)
 \end{aligned}$$

and, for example

$$\begin{aligned}
 \frac{\partial Q_{\Delta \mathbf{p}}\{f_T(\mathbf{x})\}}{\partial \Delta x} &= e^{\Delta \gamma} \left(\frac{\partial f_T(\mathbf{u})}{\partial u_1} \Big|_{\mathbf{u}=Q_{\Delta \mathbf{p}}(\mathbf{x})} \right. \\
 &\quad \times \frac{\partial u_1}{\partial \Delta x} \Big|_{u_1=(1+\Delta a_{11})(x+\Delta x)+\Delta a_{12}(y+\Delta y)+\Delta a_{13}(z+\Delta z)} \\
 &\quad + \frac{\partial f_T(\mathbf{u})}{\partial u_2} \Big|_{\mathbf{u}=Q_{\Delta \mathbf{p}}(\mathbf{x})} \\
 &\quad \times \frac{\partial u_2}{\partial \Delta x} \Big|_{u_2=\Delta a_{21}(x+\Delta x)+(1+\Delta a_{22})(y+\Delta y)+\Delta a_{23}(z+\Delta z)} \\
 &\quad + \frac{\partial f_T(\mathbf{u})}{\partial u_3} \Big|_{\mathbf{u}=Q_{\Delta \mathbf{p}}(\mathbf{x})} \\
 &\quad \times \left. \frac{\partial u_3}{\partial \Delta x} \Big|_{u_3=\Delta a_{31}(x+\Delta x)+\Delta a_{32}(y+\Delta y)+(1+\Delta a_{33})(z+\Delta z)} \right). \quad (A.4)
 \end{aligned}$$

Remembering that $\Delta \mathbf{p} = \mathbf{0}$, the previous expression simplifies into

$$\frac{\partial Q_{\Delta \mathbf{p}}\{f_T(\mathbf{x})\}}{\partial \Delta x} = \frac{\partial f_T(\mathbf{x})}{\partial x}. \quad (A.5)$$

We then proceed similarly for the other components. Finally, the algorithm reads as follows.

- 1) The initial condition is $\mathbf{p} = \mathbf{p}_0$, $\Delta \mathbf{p} = \mathbf{0}$.
- 2) Compute the inverse transformation of the reference image $Q_{\mathbf{p}^{-1}}\{f_R(\mathbf{x})\}$. Compute the resulting mean-square error using (22). Test for convergence; if reached, then quit; else, continue at step 3.
- 3) Test for a reduction of ε^2 . If reduced, then do step 3.1; else, do step 3.2.
 - 3.1) Decrease λ and update $\mathbf{p}_0 = \mathbf{p}$.
 - 3.2) Increase λ .
- 4) Solve (14) for $\Delta \mathbf{p}$. According to (23), update $\mathbf{p} = \mathbf{p}_0 \circ \Delta \mathbf{p} = (\gamma + \Delta \gamma, \mathbf{A}(\mathbf{I} + \Delta \mathbf{A}), \mathbf{b} + \mathbf{A}(\mathbf{I} + \Delta \mathbf{A})\Delta \mathbf{b})^\top$. Continue at step 2.

ACKNOWLEDGMENT

The authors thank the members of the In Vivo NMR Research Center and the National Institute on Alcohol Abuse and Alcoholism, National Institutes of Health, for providing many experimental data sets and useful discussions. Thanks also to M. Eden for perceptive comments and criticisms.

REFERENCES

- [1] L. G. Brown, "A survey of image registration techniques," *ACM Comput. Surv.*, vol. 24, pp. 325–376, Dec. 1992.
- [2] J. V. Chapnick *et al.*, "Techniques for multimodality image registration," in *Proc. Ann. Northeast Bioengineering Conf.*, New York, 1993, vol. 19, pp. 221–222.
- [3] P. A. Van den Elsen, E.-J. D. Pol, and M. A. Viergever, "Medical image matching—A review with classification," *IEEE Eng. Med. Biol. Mag.*, vol. 12, pp. 26–39, Mar. 1993.
- [4] D. W. Marquardt, "An algorithm for least-squares estimation of non-linear parameters," *J. Soc. Indust. Appl. Math.*, vol. 11, pp. 431–441, June 1963.
- [5] J. V. Hajnal *et al.*, "A registration and interpolation procedure for subvoxel matching of serially acquired MR images," *J. Comput. Assist. Tomogr.*, vol. 19, pp. 289–296, Mar. 1995.
- [6] M. Unser, A. Aldroubi, and C. R. Gerfen, "A multiresolution image registration procedure using spline pyramids," in *Proc. SPIE, Mathematical Imaging: Wavelet Applications in Signal and Image Processing*, San Diego, CA, July 15–16, 1993, vol. 2034, pp. 160–170.
- [7] A. Aldroubi, M. Unser, and M. Eden, "Cardinal spline filters: Stability and convergence to the ideal sinc interpolator," *Signal Process.*, vol. 28, pp. 127–138, 1992.
- [8] P. Gerlot-Chiron and Y. J. C. Bizais, "Registration of multimodality medical images using a region overlap criterion," *Comput. Vis., Graph., Image Process.: Graphic. Models Image Process.*, vol. 54, pp. 396–406, Sept. 1992.
- [9] A. B. Abche, G. S. Tzanakos, and E. Micheli-Tzanakou, "Effect of the spatial resolution on the precision of parameter estimation in 3-D image registration with external markers," in *Proc. 16th Ann. Int. Conf. IEEE Engineering in Medicine and Biology Soc.*, Baltimore, MD, Nov. 3–6, 1994, vol. 1, pp. 690–691.
- [10] —, "Study of the precision of registration in 3-D multimodality imaging with external markers using simulated images," in *Proc. 16th Ann. Int. Conf. IEEE Engineering in Medicine and Biology Soc.*, Baltimore, MD, Nov. 3–6, 1994, vol. 1, pp. 622–623.
- [11] Q. Zheng and R. Chellappa, "A computational vision approach to image registration," *IEEE Trans. Image Processing*, vol. 2, pp. 311–326, July 1993.
- [12] T. Wang *et al.*, "Interactive identification of marker position on 3D CT-based model," in *Proc. 16th Ann. Int. Conf. IEEE Engineering in Medicine and Biology Soc.*, Baltimore, MD, Nov. 3–6, 1994, vol. 1, pp. 530–531.
- [13] R. P. Woods, S. R. Cherry, and J. C. Mazziotta, "Rapid automated algorithm for aligning and reslicing PET images," *J. Comput. Assist. Tomogr.*, vol. 16, pp. 620–633, July/Aug. 1992.
- [14] J. Y. Chiang and B. J. Sullivan, "Coincident bit counting—A new criterion for image registration," *IEEE Trans. Med. Imag.*, vol. 12, pp. 30–38, Mar. 1993.
- [15] M. Herbin *et al.*, "Automated registration of dissimilar images: Application to medical imagery," *Comput. Vis., Graph., Image Process.*, vol. 47, pp. 77–88, July 1989.
- [16] A. Venot, L. Pronzato, and E. Walter, "Comments about the coincident bit counting (CBC) criterion for image registration," *IEEE Trans. Med. Imag.*, vol. 13, pp. 565–566, Sept. 1994.
- [17] Y. Amit, "A nonlinear variational problem for image matching," *SIAM J. Sci. Comput.*, vol. 15, pp. 207–224, Jan. 1994.
- [18] R. Szeliski and S. Lavallée, "Matching 3-D anatomical surfaces with nonrigid deformations using octree-splines," in *Proc. IEEE Workshop on Biomedical Image Analysis*, Seattle, WA, June 1994, pp. 144–153.
- [19] J. Flusser, "An adaptive method for image registration," *Pattern Recognit.*, vol. 25, pp. 45–54, Jan. 1992.
- [20] K. Matsui, M. Sase, and Y. Kosugi, "Medical image mapping using collaborative genetic algorithm," in *Proc. 16th Ann. Int. Conf. IEEE Engineering in Medicine and Biology Soc.*, Baltimore, MD, Nov. 3–6, 1994, vol. 1, pp. 612–613.
- [21] G. E. Christensen *et al.*, "Individualizing neuroanatomical atlases using a massively parallel computer," *Computer*, vol. 29, pp. 32–38, Jan. 1996.
- [22] S. Sandor and R. Leahy, "Automatic anatomical labeling of magnetic resonance brain images," in *Proc. 16th Ann. Int. Conf. IEEE Engineering in Medicine and Biology Soc.*, Baltimore, MD, Nov. 3–6, 1994, vol. 1, pp. 553–554.
- [23] B. C. S. Tom, S. N. Efstratiadis, and A. K. Katsaggelos, "Motion estimation of skeletonized angiographic images using elastic registration," *IEEE Trans. Med. Imag.*, vol. 13, pp. 450–460, Sept. 1994.
- [24] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 239–256, Feb. 1992.
- [25] D. L. G. Hill and D. J. Hawkes, "Medical image registration using knowledge of adjacency of anatomical structures," *Image Vis. Comput.*, vol. 12, pp. 173–178, Apr. 1994.
- [26] E. De Castro and C. Morandi, "Registration of translated and rotated images using finite fourier transforms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, pp. 700–703, Sept. 1987.
- [27] S. P. Kim and W.-Y. Su, "Subpixel accuracy image registration by spectrum cancellation," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Minneapolis, MN, Apr. 27–30, 1993, vol. 5, pp. V-153–V-156.
- [28] V. R. Mandava, J. M. Fitzpatrick, and D. R. Pickens III, "Adaptive search space scaling in digital image registration," *IEEE Trans. Med. Imag.*, vol. 8, pp. 251–262, Sept. 1989.
- [29] J. Noack and D. Sutton, "An algorithm for the fast registration of image sequences obtained with a scanning laser ophthalmoscope," *Phys. Med. Biol.*, vol. 39, pp. 907–915, May 1994.
- [30] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," *Image Vis. Comput.*, vol. 10, pp. 145–155, Apr. 1992.
- [31] M. Irani and S. Peleg, "Improving resolution by image registration," *Comput. Vis., Graph., Image Process.: Graph. Models Image Process.*, vol. 53, pp. 231–239, May 1991.
- [32] M. D. Pritt, "Automated subpixel image registration of remotely sensed imagery," *IBM J. Res. Develop.*, vol. 38, pp. 157–166, Mar. 1994.
- [33] A. Collignon *et al.*, "Automated multi-modality image registration based on information theory," in *Information Processing in Medical Imaging*, Y. J. C. Bizais, Ed. Boston, MA: Kluwer, 1995, pp. 263–274.
- [34] P. Viola and W. M. Wells III, "Alignment by maximization of mutual information," in *Proc. Int. Conf. Comput. Vis.*, Boston, MA, June 1995, pp. 16–23.
- [35] Q. Zheng and R. Chellappa, "Motion detection in image sequences acquired from a moving platform," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, Minneapolis, MN, Apr. 27–30, 1993, vol. 5, pp. V-201–V-204.
- [36] U. E. Ruttimann, M. Unser, and D. Rio, "Statistical analysis of image differences by wavelet decompositions," in *Proc. 16th Ann. Int. Conf. IEEE Engineering in Medicine and Biology Soc.*, Baltimore, MD, Nov. 3–6, 1994, vol. 1, pp. 28a–29a.
- [37] I. J. Cox, J. B. Kruskal, and D. A. Wallach, "Predicting and estimating the accuracy of a subpixel registration algorithm," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 721–734, Aug. 1990.
- [38] A. Goshtasby, D. A. Turner, and L. A. Ackerman, "Matching of tomographic slices for interpolation," *IEEE Trans. Med. Imag.*, vol. 11, pp. 507–516, Dec. 1992.
- [39] P. Thévenaz and M. Unser, "A pyramid approach to sub-pixel image fusion based on mutual information," in *Proc. IEEE Int. Conf. Image Processing*, Lausanne, Switzerland, Sept. 16–19, 1996, vol. I, pp. 265–268.
- [40] U. E. Ruttimann, P. J. Andreason, and D. Rio, "Head motion during positron emission tomography: Is it significant?," *Psych. Res.: Neuroimag.*, vol. 61, pp. 43–51, 1995.
- [41] G. A. Korn and T. M. Korn, *Mathematical Handbook for Scientists and Engineers*. New York: McGraw-Hill, 1968.
- [42] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes*. Cambridge, U.K.: Cambridge Univ. Press, 1986.
- [43] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part II—Efficient design and applications," *IEEE Trans. Signal Processing*, vol. 41, pp. 834–848, Feb. 1993.
- [44] —, "B-spline signal processing: Part I—Theory," *IEEE Trans. Signal Processing*, vol. 41, pp. 821–832, Feb. 1993.
- [45] C. E. Shannon, "Communication in the presence of noise," in *Proc. IRE*, Jan. 1949, vol. 37, pp. 10–21.
- [46] M. Unser, A. Aldroubi, and M. Eden, "The L_2 polynomial spline pyramid," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 364–379, Apr. 1993.
- [47] N. L. Sicotte, R. P. Woods, and J. C. Mazziotta, "Automated image registration using a 105 parameter nonlinear model," in *Proc. 2nd Int. Conf. Functional Mapping of the Human Brain*, Boston, MA, June 17–21, 1996, p. S123.



Philippe Thévenaz was born in Lausanne, Switzerland, in 1962. He received the Dipl. in microtechnology from the Lausanne Swiss Federal Institute of Technology in 1986. He received Ph.D. degree in 1993, from the Institute of Microtechnology of the University of Neuchâtel, Neuchâtel, Switzerland, with a thesis on the use of the linear prediction residue for text-independent speaker recognition.

He was a Research Assistant and Scientist at the Institute of Microtechnology of the University of Neuchâtel, where he worked as research assistant and scientist, first in the domain of image processing (optical flow), then in the domain of speech processing (speech coding and speaker recognition). He was a Visiting Associate with the Biomedical Engineering and Instrumentation Program, National Center for Research Resources, National Institutes of Health, Bethesda, MD. He is currently Head Assistant with the Biomedical Imaging Group, Swiss Federal Institute of Technology, Lausanne, Switzerland. His research interests include splines and multiresolution signal representations, geometric image transformations, and biomedical image registration.



Urs E. Ruttimann received the undergraduate degree in electrical engineering from the Eidgenössische Technische Hochschule (ETH), Zürich, Switzerland in 1966, and the Ph.D. degree in biomedical engineering from the University of Pennsylvania, Philadelphia, in 1972.

In 1972, he became Assistant Professor in clinical engineering at the George Washington University School of Medicine, Washington, D.C. Since 1980, he has held academic appointments with the Department of Child Health and Development, and in 1993 became Research Professor of Pediatrics. From 1980 to present he has been a Biomedical Engineer at the National Institutes of Health, until 1991 with the National Institute of Dental Research, and presently with the National Institute on Alcohol Abuse and Alcoholism, Bethesda, MD. His research interests include biological signal analysis, and the application of multiresolution methods to 3-D medical images.

Dr. Ruttimann is a member of the IEEE Societies for Engineering in Medicine and Biology; Acoustics, Speech, and Signal Processing; and Computers, the American Association for the Advancement of Science, the SPIE International Society for Optical Engineering, and is in the *Who's Who Directory of Optical Scientists and Engineers*.



Michael Unser (M'89–SM'94) was born in Zug, Switzerland, on April 9, 1958. He received the M.S. (summa cum laude) and Ph.D. degrees in electrical engineering in 1981 and 1984, respectively, from the Swiss Federal Institute of Technology, Lausanne, Switzerland.

He has been with the Biomedical Engineering and Instrumentation Program, National Institutes of Health, Bethesda, MD, since 1985, where he was Head of the Image Processing Group. He is currently Professor of Biomedical Image Processing at the Swiss Federal Institute of Technology, Lausanne, Switzerland. His research interests include the application of image processing and pattern recognition techniques to various biomedical problems, multiresolution algorithms, wavelet transforms, and the use of splines in signal processing. He is the author of over 60 published journal papers in these areas.

Dr. Unser serves as an Associate Editor for the IEEE SIGNAL PROCESSING LETTERS. He is a former Associate Editor (1992–1995) for the IEEE TRANSACTIONS ON IMAGE PROCESSING. He is a member of the Image and Multidimensional Signal Processing Committee of the IEEE Signal Processing Society. He is also on the editorial boards of *Signal Processing* and *Pattern Recognition*. He co-organized the 1994 IEEE-EMBS Workshop on Wavelets in Medicine and Biology, and serves as conference chair for SPIE's Conference on Wavelet Applications in Signal and Image Processing, which has been held annually since 1993. He received the Dommer prize for excellence from the Swiss Federal Institute of Technology in 1981, the research prize of the Brown-Bowery Corporation, Switzerland, for his thesis in 1984, and the IEEE Signal Processing Society's 1995 Best Paper Award (IMDSP technical area) for a paper co-authored with A. Aldroubi and M. Eden on B-spline signal processing.