

Using Markers to Compensate Displacements in MRI Volume Sequences

Thierry Viéville

► **To cite this version:**

Thierry Viéville. Using Markers to Compensate Displacements in MRI Volume Sequences. [Research Report] RR-4054, INRIA. 2000. inria-00072583

HAL Id: inria-00072583

<https://hal.inria.fr/inria-00072583>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

*Using markers to compensate displacements in MRI
volume sequences.*

T. Viéville

N° 4054

Novembre 2000

THÈME 3



*Rapport
de recherche*

Using markers to compensate displacements in MRI volume sequences.

T. Viéville

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet RobotVis

Rapport de recherche n° 4054 — Novembre 2000 — 42 pages

Abstract: The present paper proposes to compensate displacements and distortions in MRI image sequences using markers attached to the skull.

We develop a multi-model, parametric method based on sub-voxelic interpolation of the known marker geometry and analyse the performances, limits and perspectives of the method.

Key-words: RMI imagery, Motion compensation, Parametric estimation.

Utilisation de marqueurs pour compenser le déplacement dans une séquence d'images volumiques d'IRM.

Résumé : Dans ce papier on se propose de compenser les déplacements et les distorsions observées au sein de séquences d'images IRM à l'aide de marqueurs fixés sur le crâne du sujet.

On développe une méthode multi-modèles, basée sur une interpolation sous-voxélique de la géométrie, connue, du marqueur utilisé. On analyse les performances, limites et perspectives de la méthode.

Mots-clés : Imagerie IRM, Compensation de Mouvement, Estimation paramétrique.

1 Introduction

Monkey's aMRI/fMRI registration.

The analysis of Magnetic Resonance Imagery (MRI) [31, 17] helps to determine cortical regions responsive to different type of stimuli, attempting to provide neuro-scientists with a better understanding of the central nervous system.

From one MRI acquisition to another, the subject head may have moved [7], depending on the perceptual task performed or simply because of the subject breath or residual muscular activity. The worst case to tackle seems to be experiments on monkeys (which this study is dedicated to) mainly because of (i) the smaller size of the brain and body and (ii) sucking movements related to the fluid reward used to motivate the subject during the experiment.

Furthermore, the brain activity is evaluated though several so called “functional” MRI (fMRI) and has to be related to the brain anatomy obtained using “anatomical” MRI, say aMRI. The obtained data volumes have thus to be aligned with a fMRI and/or aMRI reference.

In this last case, the transformation between two data volumes is not a simple rigid displacement, whereas variations in scale or even relative deformations occur [8]. Such a multi-modal image registration [24, 11] is not easily feasible for two very different reasons:

(i) the data in fMRI and aMRI acquisitions may be totally different (see Fig¹. 1), with no common structural information to relate them (e.g. the head skull is not visible in fMRI) but even worst,

(ii) scientists precisely want to analyse the spatial relationships between both modalities and not cancel them;

as a consequence, any, even powerful, method which compensate deformations between both data volumes may also cancel or bias what is to be measured.

It is thus fundamental to register such pairs of data using an *independent* source of measure.

This is the reason why, in some approach, small conic cylinders filled with a paramagnetic substance (e.g. vitamin E) are rigidly fixed upon the skull

¹Images Shown By Courtesy Of Pr. G. Orban, Laboratorium Voor Neuro- En Psychofysiologie, Leuven.

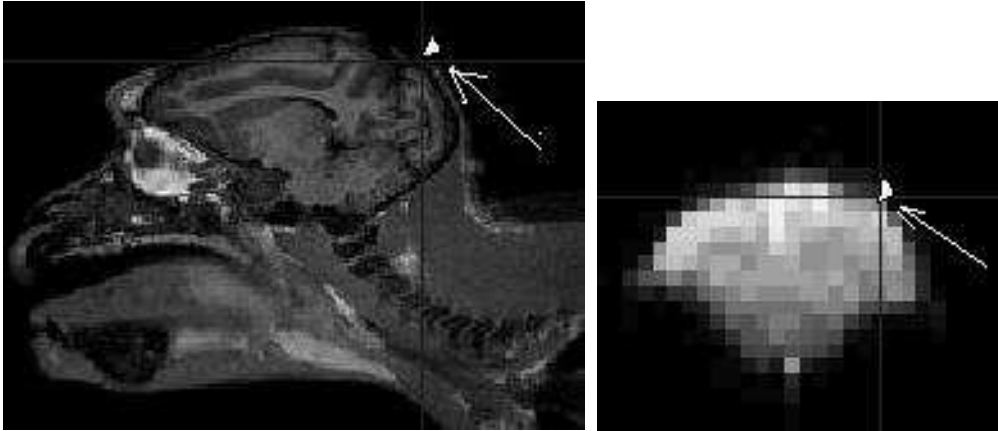


Figure 1: Schematic representation of a fiducial marker on anatomic MRI (left) or functional MRI (right) for a typical experiment on a monkey.

of the observed monkeys. Such fiducial *markers* should easily be detected as illustrated in Fig. 1.

From monkey to human aMRI/fMRI registration.

The adaptation of this approach to human MRI has also to be investigated. In such a case, the fiducial markers will not be attached (but only stick) to the skull. As a consequence, they may slip and this is taken into account by our method.

However, the present method could also be applied to a human non-invasive head-reference system² such as the one³ illustrated in Fig⁴ 3 or multi-modal radiographic markers, as illustrated in Fig 2.

Research software for auto image alignment with this “anatomark” frame is available for a research license through Brigham and Women’s Hospital.

The present developments appears as software module which may improve the performances of the current software.

²We are thankful to Jean-Baptiste Poline poline@shfj.cea.fr and Jean-François Mangin mangin@shfj.cea.fr for their precious advices regarding this extension of the study.

³This device is realized by <http://www.anatomark.com/>

⁴ Images shown by courtesy of AnatoMark, from Interneuron Pharmaceuticals, Inc.

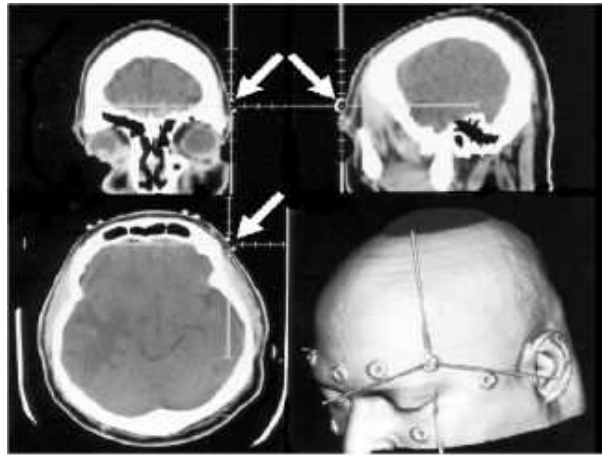


Figure 2: Multi-modal radiographic markers in human, using head locations with a minimal skin slip.

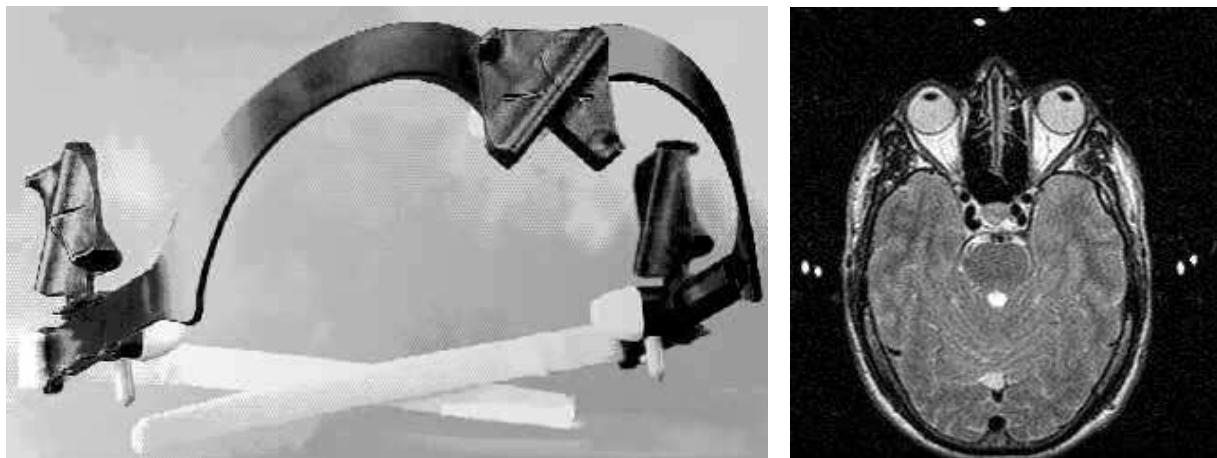


Figure 3: A human head reference system. *Left view*: the mechanic of the system. *Right view*: an example of aMRI with the markers visible as white blobs.

What is the paper about.

Following this track, in the present paper, we propose a robust and precise method to:

1. interactively locate such a fiducial marker in MRI data volumes and track them in a fMRI volumes sequence,
2. compute from these markers the transformation from a data volume to another,
 - taking into account sub-voxel accuracy and the fact that some markers may have slipped and
 - using a local affine model of transformation.

2 Setting the equations.

2.1 Models of transformation.

Parameterisation by affine transformations.

We consider that an affine transformation relates the two data volumes. This, roughly speaking, corresponds to a *rigid displacement*, *variation in scales* and some *linear distortions* or “shearing”.

More precisely⁵, two corresponding points \mathbf{M} and \mathbf{M}' in the data volumes are related by:

$$\mathbf{M}' = \underbrace{[\mathbf{S}(\lambda, \mathbf{s}) \mathbf{R}(\mathbf{r})]}_{\mathbf{A}} \mathbf{M} + \mathbf{t} \quad (1)$$

where $\mathbf{R}(\mathbf{r})$ is a 3D-rotation (i.e. an orthogonal matrix) while $\mathbf{S}(\lambda, \mathbf{s})$ is a symmetric matrix written:

$$\mathbf{S}(\lambda, \mathbf{s}) = \begin{bmatrix} 1 - \lambda_0 - \lambda_1 - \lambda_2 & s_2 & s_1 \\ s_2 & 1 - \lambda_0 - \lambda_1 + \lambda_2 & s_0 \\ s_1 & s_0 & 1 - \lambda_0 + \lambda_1 \end{bmatrix} \quad (2)$$

with $\lambda = (\lambda_0, \lambda_1, \lambda_2)^T$. Here we have chosen, because of the MRI technology, to differentiate a global scale λ_0 , a differential scale λ_1 in the “slice” direction (here the Z axis) and a differential scale λ_2 in the slice plane. Furthermore, $\mathbf{s} = (s_0, s_1, s_2)^T$ represents the shearing (i.e. linear distortions) parameters.

⁵**Notations:** We write vectors and matrices in bold letters, matrices being written with capital letters. The duals of vectors are represented as the transpose of a vector and scalars in italic, the dot-product being written as $\mathbf{x}^T \mathbf{y}$ and the cross-product $\mathbf{x} \times \mathbf{y}$ or $[\mathbf{x}]_{\times} \mathbf{y}$. The identity matrix is written \mathbf{I} .

This decomposition $\mathbf{A} = \mathbf{S}\mathbf{R}$ is in direct relation with the singular value decomposition $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ [10] of the matrix \mathbf{A} (here \mathbf{U} and \mathbf{V} are orthogonal matrices and \mathbf{D} a diagonal matrix) writing $\mathbf{S} = \mathbf{U}\mathbf{D}\mathbf{U}^T$ and $\mathbf{R} = \mathbf{U}\mathbf{V}^T$.

As a consequence, the affine transformation is parameterised by a vector \mathbf{q} :

$$\mathbf{q} = [\mathbf{t}, \mathbf{r}, \lambda, \mathbf{s}] \quad (3)$$

Parameterising 3D rotations.

Here, we represent a 3D-rotation $\mathbf{R}(\mathbf{r})$ (and also $\mathbf{S}(\mathbf{s})$) of angle $\theta \in [0, \pi[$ around an axis of direction \mathbf{u} by the vector $\mathbf{r} = 2 \tan(\frac{\theta}{2}) \frac{\mathbf{u}}{\|\mathbf{u}\|} = \theta \frac{\mathbf{u}}{\|\mathbf{u}\|} + o(\theta^3)$.

Hence, the rotation matrix is given by a variant⁶ of the Cayley [12] and Rodriguez [25] formulas, as a rational function of \mathbf{r} :

$$\mathbf{R}(\mathbf{r}) = \left(\mathbf{I} - \frac{[\mathbf{r}]_{\times}}{2} \right)^{-1} \left(\mathbf{I} + \frac{[\mathbf{r}]_{\times}}{2} \right) = \mathbf{I} + \frac{[\mathbf{r}]_{\times} + \frac{1}{2} [\mathbf{r}]_{\times}^2}{1 + \frac{\mathbf{r}^T \mathbf{r}}{4}} = \mathbf{I} + [\mathbf{r}]_{\times} + \frac{1}{2} [\mathbf{r}]_{\times}^2 + o(\|\mathbf{r}\|^3) \quad (4)$$

with the notation⁷ $[\mathbf{r}]_{\times} \mathbf{x} = \mathbf{r} \times \mathbf{x}$, while on the reverse, given a rotation matrix \mathbf{R} the related rotation vector is easily obtain:

$$\mathbf{r} = \frac{4}{1+t} \mathbf{v} \text{ with } [\mathbf{v}]_{\times} = \frac{\mathbf{R} - \mathbf{R}^T}{2} \text{ and } t = \text{trace}(\mathbf{R}) = 1 + 2 \cos(\theta) > -1 \quad (5)$$

This is well-defined for all rotations $\theta \neq \Pi$. For rotations with $\theta = \Pi$, i.e. symmetries, the limit of this formula $\mathbf{R} = \mathbf{I} + \frac{2[\mathbf{u}]_{\times}^2}{\|\mathbf{u}\|^2}$ is valid.

First and second order expansion of the rotation matrix are directly given by the previous formula, thanks to our choice of representation, while we can write:

$$\partial \mathbf{R}(\mathbf{r}) = \left(\mathbf{I} - \frac{[\mathbf{r}]_{\times}}{2} \right)^{-1} \partial [\mathbf{r}]_{\times} \left(\mathbf{I} - \frac{[\mathbf{r}]_{\times}}{2} \right)^{-1} = \partial [\mathbf{r}]_{\times} + o(\|\mathbf{r}\|^2) \quad (6)$$

⁶This representation is equivalent of using unary quaternion of the form $\mathbf{q} = [\cos(\frac{\theta}{2}) | \sin(\frac{\theta}{2}) \frac{\mathbf{u}}{\|\mathbf{u}\|}]^T$, writing \mathbf{r} in homogeneous coordinates.

⁷Thus $[\mathbf{r}]_{\times} = \begin{bmatrix} 0 & -r_2 & r_1 \\ r_2 & 0 & -r_0 \\ -r_1 & r_0 & 0 \end{bmatrix}$ with $\mathbf{r} = (r_0, r_1, r_2)^T$.

In other words $\frac{\partial \mathbf{R}(\mathbf{r})_{ij}}{\partial \mathbf{r}^k} = \varepsilon_{ijk} + o(\theta)$ where ε_{ijk} is the Eddington (or Levi-Civita)

symbol is defined by the following relations: $\varepsilon_{ijk} = \begin{cases} 0 & \text{if } i=j & \text{or } j=k & \text{or } i=k \\ 1 & \text{if } i < j < k & \text{or } j < k < i & \text{or } k < i < j \\ -1 & \text{if } j < i < k & \text{or } i < k < j & \text{or } k < j < i \end{cases}$

Given a set of N vector couples $\{(\mathbf{p}_1, \mathbf{q}_1), (\mathbf{p}_2, \mathbf{q}_2), \dots, (\mathbf{p}_N, \mathbf{q}_N)\}$ such that $\mathbf{q}_i \simeq \mathbf{R}(\mathbf{r}) \mathbf{p}_i$, with $\|\mathbf{p}_i\| = \|\mathbf{q}_i\| = 1$, we may estimate the related rotation vector using (e.g. [6]) the criterion:

$$\mathcal{L}_{\mathbf{r}} = \sum_{i=1}^N \|\mathbf{R}(\mathbf{r}) \mathbf{p}_i - \mathbf{q}_i\|^2 = \sum_{i=1}^N 2(1 - \cos(\alpha_i)) = \sum_{i=1}^N \alpha_i^2 + o(\alpha_i^4) \quad (7)$$

with $\alpha_i = \mathbf{q}_i \widehat{\mathbf{R}(\mathbf{r})} \mathbf{p}_i$, which may be written:

$$\mathcal{L}_{\mathbf{r}} = \begin{pmatrix} \mathbf{s} \\ c \end{pmatrix}^T \underbrace{\sum_{i=1}^N \begin{pmatrix} \|\mathbf{p}_i + \mathbf{q}_i\|^2 \mathbf{I} - 2(\mathbf{p}_i \mathbf{q}_i^T + \mathbf{q}_i \mathbf{p}_i^T) & 2(\mathbf{p}_i \wedge \mathbf{q}_i) \\ 2(\mathbf{p}_i \wedge \mathbf{q}_i)^T & \|\mathbf{p}_i - \mathbf{q}_i\|^2 \end{pmatrix}}_{\mathbf{M}} \begin{pmatrix} \mathbf{s} \\ c \end{pmatrix} \quad (8)$$

while $c = \cos(\frac{\theta}{2}) = \frac{1}{\sqrt{1 + \mathbf{r}^T \mathbf{r}/4}}$ and $\mathbf{s} = \sin(\frac{\theta}{2}) \frac{\mathbf{u}}{\|\mathbf{u}\|} = \frac{c}{2} \mathbf{r}$ so that $\mathbf{r} = \frac{2}{c} \mathbf{s}$.

We thus can estimate the rotation vector \mathbf{r} though (\mathbf{s}, c) as a “linear quadratic problem” since the optimal value for (\mathbf{s}, c) corresponds to the eigen-vector of \mathbf{M} associated with the smallest eigen-value.

Using a hierarchy of models.

The key point here is that we can explicit a hierarchy of transformations corresponding to different practical situations, i.e.:

	Parameter	Dimension
No displacement		0
Pure translation	\mathbf{t}	3
Rigid displacement	\mathbf{t}, \mathbf{r}	6
Scaled translation	\mathbf{t}, λ	6
Scaled displacement	$\mathbf{t}, \mathbf{r}, \lambda$	9
Affine transformation	$\mathbf{t}, \mathbf{r}, \lambda, \mathbf{s}$	12

and attempt to estimate a model with a *minimal number of significant parameters*, i.e. of “degrees of freedom”, using a more “general” model (as schematized in Fig. 4) if and only if its statistical significance is higher.

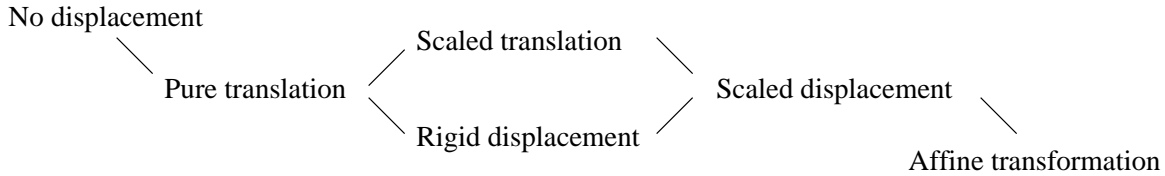


Figure 4: The model hierarchy to estimate a global displacement.

A specific model is simply specified by the fact that some components of the displacement parameter \mathbf{q} defined in (3) vanishes, which is very easy to manage. A step further, we also consider for of numerical stability reasons, 1st and 2nd order expansions of the rotation matrix as defined in (4). Furthermore, for each 3D-vector $(\mathbf{t}, \mathbf{r}, \lambda, \mathbf{s})$ we also check if some its components could be zero. This is due to the fact that MRI data is made of “slices” and that the X and Y are not homogeneous with respect to the Z direction. This finally defines up to $3 \cdot 2^{12} = 12288$ models although, of course, we will only use a realistic sub-set of 27 models made explicit in table 2. As discussed in [27] for another estimation problem and at a more theoretical level in [28], such an estimation module can easily be generated without explicitly testing *all* configurations, assuming a certain monotonicity for the estimation criterion. This is discussed in section 3.

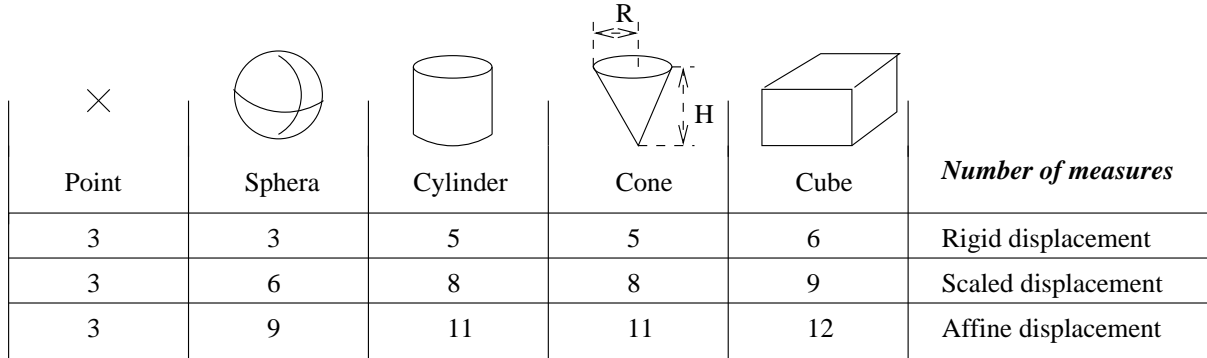
This methodology could also be generalised to non-affine transformations involving, say, quadratic terms relating two points, but the MRI physics [4] suggests that such a correction is not necessary.

In our case we will rather provide a *local affine description*, as defined in the sequel.

Designing the marker geometry.

Such markers may have different forms as illustrated in Fig. 5. Obviously, the form of the marker has an influence on the number of effective measures provided to estimate a given displacement model. For instance a spherical marker does not constrain the rotational part of the displacement whereas a cylindrical one partially does.

From a theoretical point of view [21], the number of equations obtained from a given marker for a given displacement is related to its *invariant* with respect to the considered displacement.





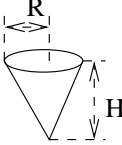
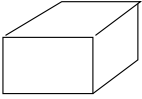
×					<i>Number of measures</i>
Point	Sphera	Cylinder	Cone	Cube	
3	3	5	5	6	Rigid displacement
3	6	8	8	9	Scaled displacement
3	9	11	11	12	Affine displacement

Figure 5: A few examples of marker geometric forms (see text for details).

In Fig. 5 we have reported (derivations, done in maple are not reported here, but left to the reader) the different numbers of measures we can expect in the general case, from a given marker geometry, given a transformation. This small set of numbers clearly shows that we better do not use (or do not model) markers as neither punctual nor spherical elements but should use cubic, cylindric or conic ones.

In particular, we will experiment the method with **conic markers**. This geometry is imposed by the available hardware.

2.2 Measurement of the marker in the data volume.

Model of the marker occurrence in the data.

As illustrated in Fig. 1 the markers are characterised by the *occurrence of a small volume of constant high (but unknown) intensity*. As a consequence, the information *inside* the volume is almost constant, while the information *outside* is random (with respect to the marker itself), in both cases not very informative. As a consequence, we have to consider cues related to *the border of fiducial marker where the intensity variation is expected to be maximal*, as

illustrated in Fig. 6.A. Such an idea has already been used in another context (e.g. [23, 9]).

More precisely, let us consider:

- (i) a given displacement parameter \mathbf{q} and
- (ii) a marker *oriented* surface distance function $d(\mathbf{V}') = 0$ which corresponds to the predefined marker surface geometry, as detailed in appendix A for our conic markers.

From (1) and (3) a point is *inside the marker* if and only if:

$$d(\mathbf{V}') < 0 \text{ with } \mathbf{V}' = \mathbf{A}(\mathbf{q}) \mathbf{V} + \mathbf{t}(\mathbf{q}) \quad (9)$$

For a given voxel, we can thus determine whether:

- (i) all voxel points are inside the fiducial marker, if and only if⁸ $d(\mathbf{V}') < -\frac{\sqrt{3}}{2}$
- (ii) all voxel points are outside the fiducial marker, if and only if $\sqrt{3} > d(\mathbf{V}') > \frac{\sqrt{3}}{2}$ or
- (iii) otherwise, the voxel contains a part of the marker border.

In class (ii) we only consider voxel points outside the marker but at a distance of less than 2 voxels to the surface border.

These three classes define a *neighbourhood* of the marker.

Implicitly, here, we assume that *the marker border corresponds to data point which intensity variation is maximal*, as illustrated in Fig. 6.A. We could have designed a method based on finding a displacement parameter for which the intensity gradient is maximal at the marker border. Such a mechanism would require (i) the explicit computation of the intensity gradient magnitude which must be done after a given and somehow arbitrary smoothing because of the noise in the data, (ii) a sampling of the marker border model in function of the data, both problems being avoided by what is proposed now.

⁸The quantity $d_0 = \frac{\sqrt{3}}{2} \simeq 0.866$ is the maximal “diagonal” distance from the voxel centre to the vertices.

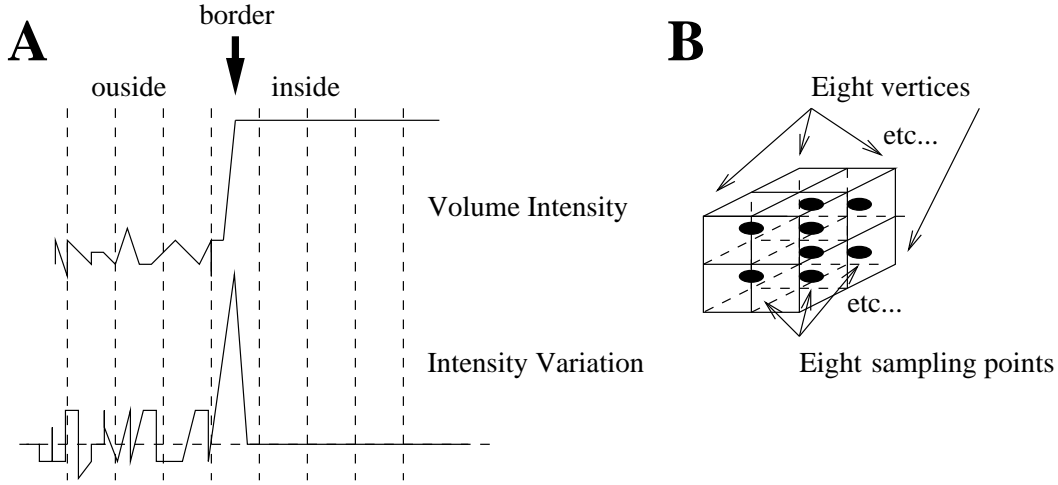


Figure 6: **A** Illustrating the cue used to locate markers. **B** Tesselation of a voxel.

Using a simple voxel intensity formation.

Here, we work at sub-voxel accuracy. In order to do so, we consider the intensity inside a voxel, as being of the form:

$$I(\mathbf{M}) = \int_{V_{oxel}} I(\mathbf{V}') d\mathbf{V}' \text{ with } I(\mathbf{V}') = \begin{cases} I_i & \text{inside the marker} \\ \mathcal{N}(M_o, V_o) & \text{outside the marker} \end{cases} \quad (10)$$

assuming a constant intensity I_i inside a marker and a normal random intensity $\mathcal{N}(M_o, V_o)$ of mean M_o and variance V_o , which represents the MRI data around the marker.

Given a neighbourhood around a marker we can thus easily estimate:

- the constant intensity I_i by averaging the intensity of all voxel of class (i) and ,
- the mean M_o and variance V_o by computing 1st and 2nd order momentum of the intensity for each voxel outside the marker.

From this estimation, we can predict from (9) and (10) the voxel intensity for points of the border, i.e. of class (iii). Here, we use a sub-voxel sampling

and define the *measurement error* :

$$v_{\mathbf{q}}(\mathbf{M}) = I(\mathbf{M}) - \frac{1}{8} \sum_{i \in \text{VoxelSamples}} I(\mathbf{V}'_i) \quad (11)$$

on 8 points inside the voxel as shown in Fig. 6.B.

Quantifying the precision of a measure.

How can we relate this *measurement error* on the intensity to the marker *displacement error* we want to correct ?

Very simply, as schematized in Fig. 7.B, up to the first order, the intensity variation is related to this displacement in proportion of the intensity slope or *gradient* in the normal direction, written with $\mathbf{g}(\mathbf{M})^T = \left. \frac{\partial I(\mathbf{V})}{\partial \mathbf{V}} \right|_{\mathbf{V}=\mathbf{M}}$.

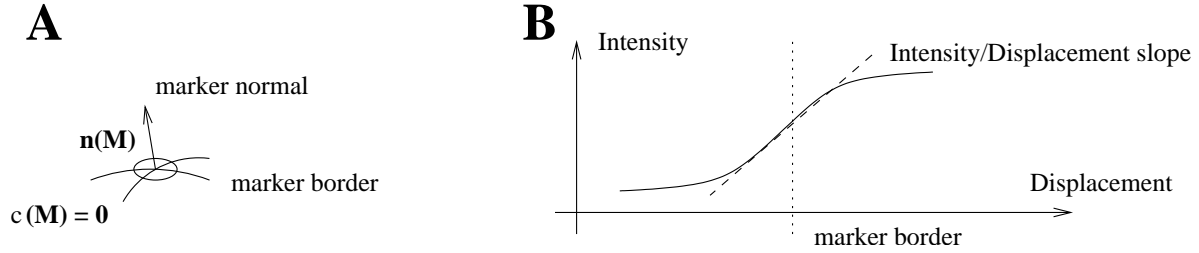


Figure 7: Quantifying the precision of a local measure. **A** Local geometry of the marker border. **B** Intensity profile at the marker border.

Furthermore, as illustrated in Fig. 7.A, it appears that, for a given point \mathbf{M} , this *local* measurement error corresponds to displacements in a direction normal to the local surface of the marker border, written $\mathbf{n}(\mathbf{M}) \equiv \left. \frac{\partial c(\mathbf{V})}{\partial \mathbf{V}} \right|_{\mathbf{V}=\mathbf{M}}$ with $\|\mathbf{n}(\mathbf{M})\| = 1$, i.e.:

$$\delta_v = \mathbf{g}(\mathbf{M})^T \left[\mathbf{n}(\mathbf{M}) \mathbf{n}(\mathbf{M})^T \right] \delta_{\mathbf{M}}$$

because $\mathbf{N} = \mathbf{n}(\mathbf{M}) \mathbf{n}(\mathbf{M})^T$ is the projection on the axis of direction $\mathbf{n}(\mathbf{M})$ (since $\mathbf{v} \parallel \mathbf{n} \Rightarrow \mathbf{N} \mathbf{v} = \mathbf{v}$) in an orthogonal direction (since $\mathbf{v} \perp \mathbf{n} \Rightarrow \mathbf{N} \mathbf{v} = 0$).

In other words, $\delta(\mathbf{M})$ is estimated in the direction $\mathbf{n}(\mathbf{M})$ in proportion of the intensity variation ($\mathbf{n}(\mathbf{M})^T \mathbf{g}(\mathbf{M})$).

In practice, we may simplify this general model, because the marker appears as a “white” blob inside a “darker” volume. We thus may assume that the gradient is approximately parallel to the surface normal and estimate its average magnitude as $\|\bar{\mathbf{g}}\| = I_i - M_o$ with a variance V_o , thus more or less constant since the variation of the intensity outside the marker appears to be negligible with respect to the intensity difference between inside and outside the marker.

As a consequence, we simply write $\delta_v = \|\bar{\mathbf{g}}\| \mathbf{n}(\mathbf{M})^T \delta_{\mathbf{M}}$.

A step further, from (1), we easily relate $\delta_{\mathbf{M}}$ to the displacement parameter variation $\delta_{\mathbf{q}}$, since :

$$\delta_{\mathbf{M}} = \left[\frac{\partial \mathbf{A}(\mathbf{q})}{\partial \mathbf{q}} \mathbf{M} + \frac{\partial \mathbf{t}(\mathbf{q})}{\partial \mathbf{q}} \right] \delta_{\mathbf{q}}$$

We thus can compute, up to the first order, the *information* matrix (e.g. [16]), i.e. the inverse of the covariance⁹ :

$$\Lambda_{\mathbf{q}}^{-1} = \frac{\|\bar{\mathbf{g}}\|^2}{V_I} \int_S \mathbf{v}(\mathbf{M}) \mathbf{v}(\mathbf{M})^T d\mathbf{M} \quad (12)$$

with

$$\mathbf{v}(\mathbf{M})^T = \mathbf{n}(\mathbf{M})^T \left[\frac{\partial \mathbf{A}(\mathbf{q})}{\partial \mathbf{q}} \mathbf{M} + \frac{\partial \mathbf{t}(\mathbf{q})}{\partial \mathbf{q}} \right]$$

where V_I is the intensity variance. In practice $V_I \simeq 1$ since related to the quantification (thus with an error of ± 1).

With such an approach the covariance matrix is, up to the scale factor $\frac{\|\bar{\mathbf{g}}\|^2}{V_I}$, only a *function of the marker geometry*. It is thus very easy to pre-compute this matrix. For the conic marker used in our experimental paradigm, for instance, the corresponding covariance matrix is given, in (23), in appendix A.

We thus have for each marker, a quantitative estimation of its influence on the parameter to estimate.

⁹We make use of the following [15] simple relation, in which V_I is the intensity variance :
 $\mathbf{y} = \mathbf{f}^T \mathbf{x} \Rightarrow \Lambda_{\mathbf{x}}^{-1} = \mathbf{f} \Lambda_{\mathbf{y}}^{-1} \mathbf{f}^T$

3 Optimising with respect to “physical” parameters.

Position of the problem.

We have to tackle, as for many other estimation problems, the problem of non-linear implicit and approximate equations local resolution. We implement such a local estimation problem as the minimisation of a criterion.

It appears that an apparently naive, but specific method based on a precise specification of the physical parameters involved in the process may be much more efficient than general sophisticated minimisation algorithms.

As introduced in [9], the key point is to represent a parameterised object through a vector of parameters which variation is to be done step by step, which is a suitable strategy for local estimations, adaptations to limited range variations from a default value, interactive estimation where a user given initial estimate is to be refined, tracking tasks, ...

More precisely, we propose that, at the *specification level*, each numerical value x_i of the parameter to estimate

- (i) is *bounded* i.e. $x_i^{min} \leq x_i \leq x_i^{max}$ and
- (ii) has a *finite precision* $x_i^\epsilon \ll x_i^{max} - x_i^{min}$ i.e. $\forall \epsilon_i, |\epsilon_i| < x_i^\epsilon, x_i \equiv x_i + \epsilon_i$.

Aside these specifications, we will attach to such a “physical” parameter: (iii) a *default value* x^0 plus a (iv) *name* and a (v) *physical unit* (second, pixel, ...). A natural default value is:

$$x^0 = \frac{x^{min} + x^{max}}{2} \quad (13)$$

More precisely :

- two numerical values of a parameter x_i and x'_i can be considered *distinct* only if:

$$|x_i - x'_i| > 2 x_i^\epsilon \quad (14)$$

We have to double the value of the bound because each value may vary in a $\pm x_i^\epsilon$ range thus their difference may vary in twice this range.

- if, on the contrary, $|x_i - x'_i| \leq 2 x_i^\epsilon$ we cannot decide whether (i) these values are the same or (ii) differ by a quantity too small to be measurable. We thus may not say that they are equal, but only *indistinguishable*.

As a consequence two vectorial parameters will be said *indistinguishable* if and only each of their components are indistinguishable, i.e.:

$$\mathbf{x} \equiv \mathbf{x}' \Leftrightarrow \underbrace{\max_i \left[\frac{|x_i - x'_i|}{2 x_i^\epsilon} \right]}_{\|\mathbf{x} - \mathbf{x}'\|_\infty} \leq 1 \quad (15)$$

that is if and only if for each component their values are indistinguishable.

Therefore, we define the notion of "physical" parameters for which each numerical value of the parameter to estimate is bounded, has a finite precision, a default value, a name and a physical unit.

Quasi-static minimisation method.

With such specifications, and rather than using standard minimisation techniques which usually require a gradient evaluation at each iteration, we can design a much more specific and effective randomised algorithm.

The problem is now reformalised as¹⁰:

$$\min_{\mathbf{k} \in \mathcal{Z}^n} \mathcal{L}(\mathbf{q}_0 + \mathbf{k} \odot \delta_{\mathbf{q}})$$

where $\mathbf{q}_0 \in \mathcal{R}^n$ is an initial default value, $\delta_{\mathbf{q}}$ a fixed step vector.

At each instant we consider:

- a given value of the parameter \mathbf{q}_\bullet and its corresponding criterion value $\mathcal{L}_\bullet = \mathcal{L}(\mathbf{q}_\bullet)$,
- an estimation of the local descent direction \mathbf{d}_\bullet , which computation is discussed in the sequel.

In our case we calculate the gradient by finite differences.

As an initial value,

- (o) $\mathbf{q}_\bullet = \mathbf{q}_0$ and \mathbf{d}_\bullet is equal to a set of huge equal values.

¹⁰We write: $\mathbf{w} = \mathbf{u} \odot \mathbf{v}$ the component by component product of a vector i.e. with $w^i = u^i v^i$

At each step,

- (a) a parameter component $i \in \{1..n\}$ is randomly selected with a probability equal to:

$$p_i = \frac{|\mathbf{d}_\bullet^i|}{\sum_{i=1}^n |\mathbf{d}_\bullet^i|}$$

and writing $\mathbf{e}_i = \frac{(0 \cdots 1 \cdots 0)}{1 \cdots i \cdots n}$,

- (b) the criterion $\mathcal{L}^+ = \mathcal{L}(\mathbf{q}^+)$ is recomputed at

$$\mathbf{q}^+ = \mathbf{q}_\bullet \mp \mathbf{e}_i \odot \delta_{\mathbf{q}}$$

the sign of \mp depending of the descent direction i th component sign,

- (c) if $\mathcal{L}^+ < \mathcal{L}_\bullet$, \mathbf{q}^+ is taken as new \mathbf{q}_\bullet value, with the corresponding criterion value and the minimisation continues from step (a);

- (d) else the criterion $\mathcal{L}^- = \mathcal{L}(\mathbf{q}^-)$ is recomputed at

$$\mathbf{q}^- = \mathbf{q}_\bullet \pm \mathbf{e}_i \odot \delta_{\mathbf{q}}$$

the sign of \pm depending of the descent direction i th component sign.

- (e) if $\mathcal{L}^- < \mathcal{L}_\bullet$, \mathbf{q}^- is taken as new \mathbf{q}_\bullet value, with the corresponding criterion value and the minimisation continues from step (a);

- (f) else we set $\mathbf{d}_\bullet^i = 0$ and the minimisation continues from step (a) if $\mathbf{d}_\bullet^i \neq 0$ for at least one component i

- (g) if $\mathbf{q}_\bullet \neq \mathbf{q}_0$ the algorithm is reinitialised at step (o) with $\mathbf{q}_0 = \mathbf{q}_\bullet$

- (h) else the algorithm stops.

The chosen probability allows, in average, to decrease the criterion in proportion of the descent direction, since the probability is proportional to the gradient magnitude in each direction.

Furthermore, if for a parameter component i , in both directions, the criterion does not decrease (we have a “valley” in this direction) the gradient is set to zero and the component is no more selected.

A step ahead, since it might happen that the criterion has the form of a valley for a given parameter position \mathbf{q}_\bullet but not after a modification of \mathbf{q} , at step (g), the algorithm is re-initialised if something has been changed. As a consequence, we are sure to be at a local minimum when the algorithm stops.

If, for a given parameter a bound (minimal/maximal value) is attained the algorithm is stopped and an exception occurs.

Contrary to the original method proposed in [9] the occurrence of “valleys” as discussed before is managed in a more rudimentary way, simplifying and thus increasing the speed of the algorithm, but the general performances are similar.

Performances of the algorithm.

Obviously, this algorithm:

(i) always induces *minimal magnitude variations* of the parameter vector, since it modifies only one parameter at a time and only to a minimal value; this is very useful to find the *closest local minimum* when several minima occur; this may be written:

$$\mathbf{q}_\bullet = \underset{\{\mathbf{q}=\mathbf{q}_0+\mathbf{k}\odot\delta_{\mathbf{q}},\mathbf{k}\in\mathbb{Z}^n\}}{\operatorname{argmin}} \mathcal{L}(\mathbf{q}) + \|\mathbf{q} - \mathbf{q}_0\|_\epsilon \quad (16)$$

where $\|\mathbf{q} - \mathbf{q}_0\|_\epsilon$ denotes the semi-norm induced by the step-by step sampling of the parameter on each component, from a application point of view such a method is meaningful is each parameter component q_i has a physical meaning and is to be treated as such, whereas other methods tends to consider a parameter vector as a monolithic quantity;

(ii) *stops after a finite time* since a given parameter (1) is bounded and thus can not be modified indefinitely (2) is modified if and only if the criterion strictly decreases; as a consequence, the parameter evolutes in a finite space and never comes back to the same position it must then follow a finite path,

(iii) avoid computing the gradient at each step (but only one component of it) or avoid having to provide a analytic value of the gradient (although would be compatible with the method). As a consequence, each iteration is very fast and such algorithm should be very reactive.

This algorithm requires more specifications than usual methods of minimisation but thanks to these, no threshold and no internal parameter is to be tuned.

It is expected to be faster, as experimentally verified in a non trivial perceptual task [9], because it tries to find the best value only at the precision

(defined by $\delta_{\mathbf{q}}$) required whereas other methods may attempt to keep going even for negligible level of precisions;

Furthermore, each time the criterion is evaluated, the algorithm takes advantage of any observed decrease; this might be suitable if very close to the minima or when the criterion is very flat, in comparison to much heavier methods which compute 1st or 2nd order derivatives [22] at each step and may be slow down at this stage.

A step ahead, concurrent methods might have some oscillatory behaviours around the minimum [22], especially if the criterion is irregular, which is the case when considering real data that this trivial algorithm is not subject to.

From a theoretical point of view the proposed method is sub-optimal since it is more or less a 1-st order method (contrary to better formalised analytical methods such as described in [28]), while the gradient estimation is very approximate (since not recomputed at each step some components of the gradient may slightly differ from their actual value). It is deliberately “slow” in the sense that it modifies the parameter values step by step and does not attempt to predict the minimum value in one shot.

It is thus not to be used when large magnitudes of variation are expected but is to be *dedicated to local minimisation problems*. In such cases (e.g. [9]) it is much faster in terms of computation and also in terms of number of steps.

Applications of such an algorithm include the implementation of *tracking tasks in clustered environments* (e.g. [32, 14, 3]) especially when such a recursive dynamic estimation involves the *control of actuators* (e.g. [5, 26]) or to *refine a value provided by a user interaction*.

On the other hand it might be of great interest for *continuation methods* [20, 30] implementation.

Multi-model implementation.

In order to obtain an estimation with a *minimal number of parameters*, we not only want to minimise a given criterion with respect to a given parameter but also to verify if we could not:

- assume that a given parameter component vanishes,
- use a simpler model (e.g. a 1st or 2nd order expansion of a formula as in (4)) defined a by a qualitative variable.

Such a methodology is now of common use in some very huge problems of parameter estimation [29, 27, 18] but contrary to a very general formulation such as done in [28] here the software implementation will be much simpler and in fact much effective.

This is due to the fact that a model is not characterised by a set of implicit constraints [28] or that complex inter-dependent relations must be taken into account to avoid equivalent or incoherent models (this being undetectable at a numerical level) [18].

Here, the model hierarchy is static and has been defined “by hand”. It is given in table 2. Obviously there is somehow a “combinatory explosion” when different alternatives are combined. However, the user can (and has to !) filter among all possibilities those which are going to be really useful in practice. Following this track we have, for instance, only used the 2nd order expansion of the rotation matrix, as an intermediate computation of the general form.

$$\begin{array}{lll}
0 & \mathbf{M}' & = \mathbf{M} \\
t & \mathbf{M}' & = \mathbf{M} + \mathbf{t} \\
f & \mathbf{M}' & = (\mathbf{I} + [\mathbf{r}]_{\times}) \mathbf{M} + \mathbf{t} \\
s & \mathbf{M}' & = (\mathbf{I} + [\mathbf{r}]_{\times} + 1/2 [\mathbf{r}]_{\times}^2) \mathbf{M} + \mathbf{t} \\
r & \mathbf{M}' & = \mathbf{R}(\mathbf{r}) \mathbf{M} + \mathbf{t} \\
l & \mathbf{M}' & = \mathbf{S}(\lambda, 0) \mathbf{M} + \mathbf{t} \\
lf & \mathbf{M}' & = \mathbf{S}(\lambda, 0) (\mathbf{I} + [\mathbf{r}]_{\times}) \mathbf{M} + \mathbf{t} \\
ls & \mathbf{M}' & = \mathbf{S}(\lambda, 0) (\mathbf{I} + [\mathbf{r}]_{\times} + 1/2 [\mathbf{r}]_{\times}^2) \mathbf{M} + \mathbf{t} \\
lr & \mathbf{M}' & = \mathbf{S}(\lambda, 0) \mathbf{R}(\mathbf{r}) \mathbf{M} + \mathbf{t} \\
a & \mathbf{M}' & = \mathbf{S}(\lambda, \mathbf{s}) \mathbf{M} + \mathbf{t} \\
af & \mathbf{M}' & = \mathbf{S}(\lambda, \mathbf{s}) (\mathbf{I} + [\mathbf{r}]_{\times}) \mathbf{M} + \mathbf{t} \\
as & \mathbf{M}' & = \mathbf{S}(\lambda, \mathbf{s}) (\mathbf{I} + [\mathbf{r}]_{\times} + 1/2 [\mathbf{r}]_{\times}^2) \mathbf{M} + \mathbf{t} \\
ar & \mathbf{M}' & = \mathbf{S}(\lambda, \mathbf{s}) \mathbf{R}(\mathbf{r}) \mathbf{M} + \mathbf{t}
\end{array}$$

Table 1: Equations used in the model hierarchy, these different equations allow to (a) use 1st and 2nd order expansions and to (b) avoid computing useless parts of the equation.

With such specification, the multi-model implementation is rather simple. We may use a very simple test discussed in [28]. A model defined by a criterion $\mathcal{L}'(\mathbf{q})$ with $d' = \text{NumberOfMeasure} - \text{ParameterDimension}$ degrees of freedom is to be preferred with respect to a model $\mathcal{L}(\mathbf{q})$ with $d = \text{NumberOfMeasure} - \text{ParameterDimension}$ degrees of freedom

Label	t	r	λ	s	=	Alternatives	Name
0	0 0 0	0 0 0	0 0 0	0 0 0	0	T	No displacement
T	x x x	0 0 0	0 0 0	0 0 0	t	R1, Ts	Pure translation
R1	x x x	x x x	0 0 0	0 0 0	f	R2, Ds1	Rigid 1st-order displacement
R2	x x x	x x x	0 0 0	0 0 0	s	Rx	Rigid 2nd-order displacement
Rx	x x x	x x x	0 0 0	0 0 0	r	Dsx	Rigid displacement
Ts	x x x	0 0 0	x 0 0	0 0 0	l	Tz, Ds1	Scaled Translation
Tz	x x x	0 0 0	x x 0	0 0 0	l	Tl, Dz1	Scaled Translation
Tl	x x x	0 0 0	x x x	0 0 0	l	Dl1	Scaled Translation
Ds1	x x x	x x x	x 0 0	0 0 0	lf	Ds2, Dz1, As1	Scaled Displacement
Ds2	x x x	x x x	x 0 0	0 0 0	ls	Dsx	Scaled Displacement
Dsx	x x x	x x x	x 0 0	0 0 0	lr	Dzs, Asx	Scaled Displacement
Dz1	x x x	x x x	x x 0	0 0 0	lf	Dz2, Dl1, Az1	Scaled Displacement
Dz2	x x x	x x x	x x 0	0 0 0	ls	Dzx	Scaled Displacement
Dzx	x x x	x x x	x x 0	0 0 0	lr	Dlx, Azx	Scaled Displacement
Dl1	x x x	x x x	x x x	0 0 0	lf	Dl2, Al1	Scaled Displacement
Dl2	x x x	x x x	x x x	0 0 0	ls	Dlx	Scaled Displacement
Dlx	x x x	x x x	x x x	0 0 0	lr	Alx	Scaled Displacement
As1	x x x	x x x	x 0 0	x x x	af	Az1, As2	Affine Transformation
As2	x x x	x x x	x 0 0	x x x	as	Asx	Affine Transformation
Asx	x x x	x x x	x 0 0	x x x	ar	Azx	Affine Transformation
Az1	x x x	x x x	x x 0	x x x	af	Al1, Az2	Affine Transformation
Az2	x x x	x x x	x x 0	x x x	as	Azx	Affine Transformation
Azx	x x x	x x x	x x 0	x x x	ar	Alx	Affine Transformation
Al1	x x x	x x x	x x x	x x x	af	Al2	Affine Transformation
Al2	x x x	x x x	x x x	x x x	as	Alx	Affine Transformation
Alx	x x x	x x x	x x x	x x x	ar		Affine Transformation

Table 2: The table of the model hierarchy implemented in the localisation module. Parameter to be adjusted are denoted by a “x”. The model equation in the “=” column is given in table 1. Alternatives refer to the label given on the left.

only if:

$$\frac{\mathcal{L}'(\mathbf{q})}{d'} < \frac{\mathcal{L}(\mathbf{q})}{d}$$

for $d' > d$.

Here, the ratio $\frac{\mathcal{L}(\mathbf{q})}{d}$ is called the *cost* of the model.

It corresponds to a Fisher test with a minimal probability error [28].

Another track [2] is to quantify how much the new degree of freedom, say q_\bullet may decrease the criterion $\mathcal{L}'(\mathbf{q}, q_\bullet)$ around the optimum $\mathcal{L}(\bar{\mathbf{q}})$ obtained without considering it.

In order to avoid looking at *all alternatives* we assume that it is sufficient to look for a *local minimum* in the model hierarchy, i.e. that the cost decreases from the null-model to the optimal model (because adding new parameters

allow a better fit to the data) and then increase (because the gain in precision is not balanced by the decrease in degrees of freedom) but never re-decrease again.

Adapting the algorithm proposed in [28] to this simpler paradigm we obtain:

Initialisation Put the null-model, computing its cost, in a *candidate list* and consider it as initial *best model*.

Iteration Taking the *model* of the candidate list with *minimal cost* :

- For each alternative :
 - Estimate the model parameter and the related cost.
 - Put alternatives which cost are lower than the model cost in the candidate list, in cost increasing order.
- Remove the model from the candidate list.

Termination When the candidate list is empty.

The model of minimal cost is considered as the best model.

This algorithm has been implemented and used in this estimation module.

4 Designing the experimental paradigm.

From the previous developments, let us design an effective method to locate, track and stabilise the data using the markers. The key-point, here, is that we *consider the need of having a user interaction*. Let us discuss this point.

One may wish to design an automatic procedure, for instance considering that markers correspond to “white parts” of the data volume. One may always attempt to design a mechanism which detect the marker from such a cue, with a robustness of, say 99%; except if the marker have slipped; except if another unexpected reflection has occurred; except if yet another thing happened. Of course, one may also use contextual information (e.g. the marker should be on the monkey skin) to try to implement a way to avoid such artefacts.

On the other hand, anybody doing a MRI experiment *must take a look at the data* at least one aMRI and one fMRI. It thus can easily also take a look

at the few markers, check if they seems to be ok. This will be much *safer* and much *faster* than a huge algorithm.

As a consequence, we propose to specify that a human operator has to *click on each marker in one slice of the data volume on the aMRI and on the first fMRI of an acquisition sequence* to validate their use. From this, initial estimate we will easily design *local mechanisms* to locate and track these tokens and compute the related transformations.

Since the design of the procedure itself is not based on a formal development but some experience in similar mechanisms, let us simply describe the paradigm.

4.1 Interactive designation of the markers

1. The user clicks on one of the marker voxel in any data slide.
2. Points inside the marker are roughly detected by indexing all voxels connected to the chosen voxel and which intensity is similar (say $\pm 10\%$) to the intensity of the voxel clicked.
3. First and second order momenta of the voxel are computed so that the marker is implicitly modelised by a quadratic form of inertia of the form :

$$\begin{aligned} q(\mathbf{M}) &= (\mathbf{M} - v_1)^T \Upsilon_2 (\mathbf{M} - v_1) \\ \text{with } v_1 &= \int_{V \in \text{Marker}} V dV \\ \text{and } \Upsilon_2 &= \int_{V \in \text{Marker}} (V - v_1) (V - v_1)^T dV \end{aligned} \quad (17)$$

4. From these quantities, an initial estimation of the affine transform $\mathbf{M} = \mathbf{A} \bar{\mathbf{M}} + \mathbf{t}$ between (i) a point \mathbf{M} of the marker identified in the data volume and (ii) the corresponding point $\bar{\mathbf{M}}$ of the marker model is calculated.

Here we only estimate a rotation and translation. The other parameters being close to zero in practice, do not need an initial estimation.

In this specific case, we compute the rotation which aligned the cone axis \mathbf{z} (detected as the axis which eigen value differs from the two others, as made explicit in appendix A, eventually confirmed by an elementary analysis of the volume asymmetry) with the model cone axis $\bar{\mathbf{z}}$. It is well known that such a minimal rotation is of axis $\mathbf{z} \wedge \bar{\mathbf{z}}$ and of angle $\widehat{\mathbf{z}\bar{\mathbf{z}}}$.

More generally, from (17) we easily obtain:

$$\begin{cases} v_1 &= \mathbf{A} \bar{v}_1 + \mathbf{t} \\ \Upsilon_2 &= \mathbf{A}^T \bar{\Upsilon}_2 \mathbf{A} \end{cases} \quad (18)$$

where \bar{v}_1 and $\bar{\Upsilon}_2$ are the first and second order momenta of the quadratic form of inertia for the model.

Using the eigen-value decomposition of $\Upsilon_2 = \mathbf{R}^T \mathbf{D} \mathbf{R}$ and $\bar{\Upsilon}_2 = \bar{\mathbf{R}}^T \bar{\mathbf{D}} \bar{\mathbf{R}}$ we thus can propose, assuming $\mathbf{D} \simeq \bar{\mathbf{D}}$:

$$\begin{cases} \mathbf{A} &= \bar{\mathbf{R}}^T \mathbf{R} \\ \mathbf{t} &= v_1 - \mathbf{A} \bar{v}_1 \end{cases} \quad (19)$$

as an initial estimation of the affine transform.

From this we easily obtain the related displacement parameter \mathbf{q}_0 (see section 2.1).

5. From this initial estimate of the transformation \mathbf{q}_0 , the localisation of this marker with respect to the model is refined, minimising, on the measurement errors defined in (11), a criterion of the form:

$$\mathcal{L}_k(\mathbf{q}) = \sum_{M \in \text{Voxel Border}} \rho(v_{\mathbf{q}}(\mathbf{M})) + \|\mathbf{q} - \mathbf{q}_0\|_{\epsilon} \quad (20)$$

where $\rho(v)$ is a positive “error function” (e.g. $\rho(v) = v^2$ or $\rho(v) = |v|$) chosen to obtain a robust M-estimator [13] which tends to cancel the influence of erroneous measurement errors. Here, such artefacts may be related to the fact that a marker have slipped, but it is not expected to obtain huge erroneous values in this case. As a consequence (see [19] for details), such a criterion is expected to have good performances.

Here we use the profile discussed and develop in appendix B, for small errors the criterion behaves as a least-square criterion. Since the standard deviation on the intensity is $\sigma_I = 1$ because the uncertainty is only related to the quantification (thus its standard deviation is about 1 quantification level), the previous criterion corresponds to a Mahalanobis distance, related up to the 1st order to the likelihood of the estimate [1].

The regularisation term $\|\mathbf{q} - \mathbf{q}_0\|_\epsilon$ is defined by the minimisation mechanism which tends to maintain the estimate close to the initial estimate (see appendix 3 for details).

Furthermore, the residual value of this criterion allows to evaluate the quality of the estimation.

6. As a feedback to the user, the marker model impression is drawn in the data volume to verify, at the graphic-user interface level, that the estimation process has converged to a reasonable value.

4.2 Tracking along a data sequence.

Since we have to track these markers in several (up to a few hundred) data volume in the sequence, we want to have this step fully automatic. This is facilitated by the fact that (i) from one data volume to another the marker displacement is relatively small (although it might be larger than the marker sizes) (ii) this part of the computation may be done off-line. Furthermore, from one frame to another, (iii) the displacement is expected to be random without any correlation from one data volume to another. As illustrated in Fig. 8, this leads to the following heavy but simple heuristic.

1. Given a maximal displacement in voxel from the original location, we randomly select a new location.
2. With this location, we apply step 2 to 5 of the previous procedure, as if the new location was clicked by a user.
3. If the residual value of minimised criterion is below a given threshold we consider that the new marker location has been identified, otherwise we repeat step 1 of this procedure with a new sample.

In practice, such random a mechanism is usually more efficient than sampling all around the locations of the 3D volume.

4.3 Computation of a local affine transformation.

Thanks to the previous steps we have a localisation of each marker in each data volume.

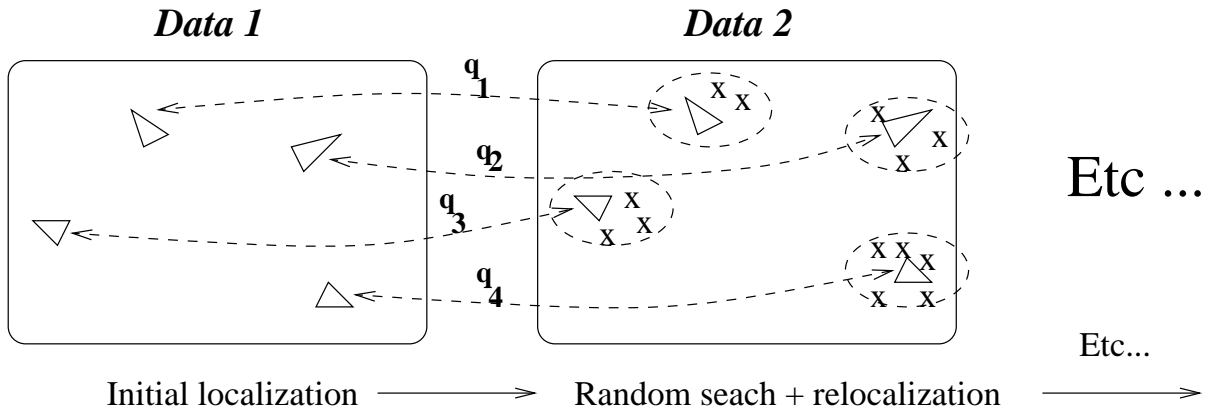


Figure 8: Trivial tracking along a fMRI sequence.

In other words, for each marker we have estimated a *separate* affine transformation.

This is useful either to avoid to bias all transformation estimates because of the slip of a marker and to help other modules to analyse the non-linear deformations in the data volume.

At the next stage, we want to compute the *relative transformation* between two data volumes. As illustrated in Fig. 9 each marker is put in correspondence (either interactively between aMRI and fMRI or automatically by tracking in fMRI image sequences).

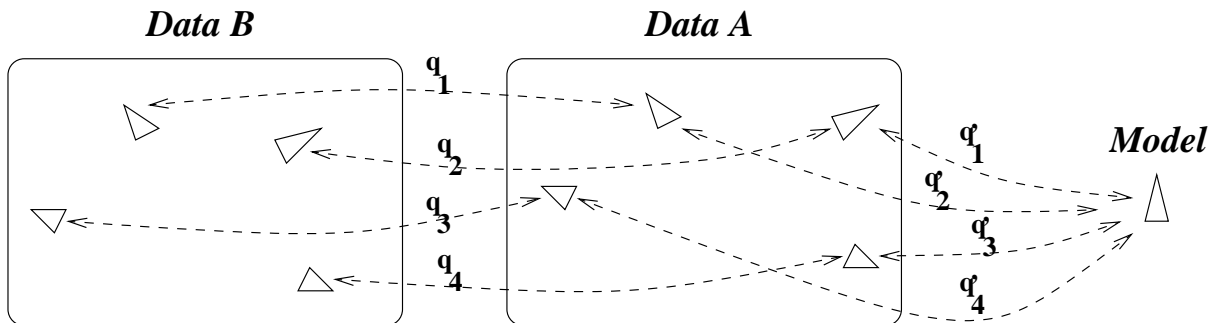


Figure 9: Combining local transformations.

We thus define the transformations between the model and the markers in one view \mathbf{q}'_k and the relative transformations \mathbf{q}_k between the markers in one view and the corresponding marker in the other view.

The transformations \mathbf{q}'' between the model and the markers in this last view is given by:

$$\begin{cases} \mathbf{M}'' = \mathbf{A}(\mathbf{q}') \mathbf{M}' + \mathbf{t}(\mathbf{q}') \\ \mathbf{M}' = \mathbf{A}(\mathbf{q}) \mathbf{M} + \mathbf{t}(\mathbf{q}) \end{cases} \Rightarrow \text{with } \begin{cases} \mathbf{M}'' = \mathbf{A}(\mathbf{q}'') \mathbf{M} + \mathbf{t}(\mathbf{q}'') \\ \mathbf{A}(\mathbf{q}'') = \mathbf{A}(\mathbf{q}') \mathbf{A}(\mathbf{q}) \\ \mathbf{t}(\mathbf{q}'') = \mathbf{A}(\mathbf{q}') \mathbf{t}(\mathbf{q}) + \mathbf{t}(\mathbf{q}') \end{cases}$$

written $\mathbf{q}'' = \mathbf{q} * \mathbf{q}'$ and easily computed from the formulas given in section 2.1.

In order to have the relative estimations \mathbf{q}_i in coherence we propose to minimise the Mahalanobis distance between each pairs, i.e. :

$$d(\mathbf{q}_{k_1}, \mathbf{q}_{k_2}) = (\mathbf{q}_{k_1} - \mathbf{q}_{k_2})^T (\mathbf{\Lambda}_{\mathbf{q}_{k_1}} + \mathbf{\Lambda}_{\mathbf{q}_{k_2}})^{-1} (\mathbf{q}_{k_1} - \mathbf{q}_{k_2}) = 2 (\mathbf{q}_{k_1} - \mathbf{q}_{k_2})^T \mathbf{\Lambda}_{\mathbf{q}}^{-1} (\mathbf{q}_{k_1} - \mathbf{q}_{k_2})$$

since in our formalism covariance matrices are the same, because they are estimated only from the model, not the data. Furthermore, since we only estimate here *relative* displacements which are expected to be of small amplitude, we compute the information matrix $\mathbf{\Lambda}_{\mathbf{q}}^{-1}$ at $\mathbf{q} \simeq 0$ in order to make use of a fixed, sparse, thus easy and fast to compute quadratic form made explicit in (23).

With these notations, our estimates can easily be combined minimising:

$$\sum_{k \in \text{Markers}}^{\text{Data A}} \mathcal{L}_k(\mathbf{q}'_k) + \sum_{k \in \text{Markers}}^{\text{Data B}} \mathcal{L}_k(\mathbf{q}_k * \mathbf{q}'_k) + 2 \sum_{k_1, k_2} (\mathbf{q}_{k_1} - \mathbf{q}_{k_2})^T \mathbf{\Lambda}_{\mathbf{q}}^{-1} (\mathbf{q}_{k_1} - \mathbf{q}_{k_2}) \quad (21)$$

This allows to:

- maintain the idea that *each marker provides an estimate of the local affine transformation of the data volume in its neighbourhood*, while
- all these estimations are put in coherence, depending on the metric induced by the Mahalanobis distance defined on the measures in (12).

It would have been however very easy to consider a *unique global affine transform* by simply defining only one parameter \mathbf{q} .

5 Experimental results.

In order to evaluate the method we have considered a true MRI sequence in which we have *artificially* introduced the marker-print, as made visible in Fig 1. Indeed, the synthetical displacement of the marker has nothing to do with the real data, but it provides a set of measure:

- which contents is entirely similar to what is to be expected in practice,
- which displacement is entirely controllable in order to evaluate the method.

We have considered a set of 4 markers (except for the 2st simulation) put on the monkey skull, and have generated rigid displacements with or without distortions. We also simulate the fact that one or two markers may have slipped.

The conic marker height is twice the radius in our case. We really need to have a marker form which is NOT symmetric to properly recover its orientation.

We have simulated the user click by randomly selecting a point inside (or sometimes outside, simulating a mistake) the marker and have located, tracked and evaluate the markers displacement.

In Fig. 10 we verify that the method offers a sub-voxelic precision. More precisely, in the case of a pure translation, the precision is directly function of the number of markers and their size, with two aspects:

- It seems we need about **four** markers to obtain optimal performances, whereas much more is not necessary (additional simulations have confirmed this point).
- The higher the marker size, the better the performances. It seems that we always gain by increasing the marker size.
- Globally, we may obtain a precision up to 0.1 voxel but the “effective” precision is really 0.3 voxel in our experiment.

This means that, with a 3mm voxel size fMRI:

using four markers with 1cm size, yields a precision better than 1mm

A step ahead, we have studied errors in orientation for a unique marker. In Fig. 11, for different rotation angles, using the definition of the rotation vector

Marker number	Marker size (voxel)				
	1	4	8	16	32
1	1.8	1.4	1.1	0.8	0.6
2	1.6	1.3	0.9	0.4	0.5
3	1.4	1.1	0.6	0.3	0.3
4	1.2	0.9	0.7	0.4	0.1
6	1.1	0.8	0.6	0.3	0.2

Figure 10: Precision in voxel for the localisation, given different marker sizes and numbers, for a pure translation.

\mathbf{r} in (4), we have computed, given an *estimated* rotation vector $\tilde{\mathbf{r}}$ and a *true* rotation vector $\bar{\mathbf{r}}$:

$$\epsilon_{\mathbf{r}} = \frac{180}{\pi} 2 \arctan\left(\frac{\|\tilde{\mathbf{r}} - \bar{\mathbf{r}}\|}{2}\right)$$

which capture errors in both orientation (although in our case, this was not really observed) and magnitude and is written in degree, to ease its interpretation. It is, up to 1st order, the angle of a rotation which could “correct” between the estimated and true value.

Remember that we always have true values here, since we *simulate the displacements within the real data*.

Remember that we estimate the values “step-by-step” (in this case a step is 0.05 deg) which explains the fact we always obtain fixed precision results.

This part of the experiment shows the limit of our “sub-voxelic” method, because our interpolation rather sensitive with respect to the *marker sampling*, i.e. for angles which preserves the symmetry of the conic pattern the precision is very high, whereas at other orientations the estimation is really biased.

True rotation angle	0	10	30	45	60	90
Error in rotation magnitude	0.05	2.90	3.20	0.15	4.30	0.05

Figure 11: Comparing true and estimated rotation magnitudes, for a unique conic marker of size 16, quantities are normalised in degree.

This is a serious drawback : it means that we can only make a limited use of the marker “deformation” to estimate the local displacement, although we would have expect so. Such a bias disappears if we increase the size of the marker to (unrealistic !) dimensions.

This may suggest to think of using a huge marker (i.e. a kind of “crown”) instead of several small ones, at least recommend to:

use markers with maximal sizes.

But we have followed another track and see, whether, in practice using another marker form could not be a better alternative (although neuro-scientists actually seem not to have this easily available, because **“conic” markers correspond to the “screw” pattern plugged in the crane**). The answer is yes :

using cubic markers limits interpolation errors,

True rotation angle	0	10	30	45	60	90
Error in rotation magnitude	0.00	0.15	1.15	0.10	1.15	0.05

Figure 12: Comparing true and estimated rotation magnitudes, for a unique cubic marker of size 16, quantities are normalised in degree.

as illustrated in Fig. 12 for the same simulation. Of course, the problem is still there but with a much smaller magnitude.

From a theoretical point of view, and aside the interpolation effects, we have tried to understand why such a result. A way to interpret this effect is to compute the covariance matrix with respect to the displacement parameters, as it was done for a conic marker, in (23). For a cubic marker, we obtain the formula given in Fig. 13 using the same assumptions as those discussed in this paper.

The key point here is that *sub-matrices related to translation parameters, rotation parameters or skew parameters are diagonal, thus invariant by rotation*. In other words, if we change the orientation, up to the 1st order, we do not modify the estimation precision. This is not the case for scales because we have intentionally distinguished inhomogeneous scales, considering slices of fMRI acquisitions. So that:

using cubic markers would have been better.

$$\Lambda_q^{-1} = \frac{\|\bar{\mathbf{g}}\|^2}{V_I} \begin{bmatrix} k_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & k_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3k_3 & k_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & k_3 & 3k_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2k_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_2 \end{bmatrix} \quad \text{with } \begin{cases} k_1 = 8S^2 \\ k_2 = 16/3S^4 \\ k_3 = 8S^4 \end{cases} \quad (22)$$

Figure 13: Information matrix for a cubic marker and a small displacement.

However, since a cube has symmetries ambiguities in orientation must be carefully taken into account and such a marker is only suitable to correct for small rotations (below 45 deg).

If we combine several markers, the precision in orientation is directly related to relative positioning of the different markers. More precisely, we have verified that the order of magnitude error in orientation ϵ_r (given now in radian) is related to the error in positioning ϵ_t and the average distance \bar{d} between markers by the expected rule: $\epsilon_r \simeq \frac{\epsilon_t}{\bar{d}}$.

We thus indeed can recommend to:

install markers on the front/back and top/bottom of the head.

Further on, we have evaluated the robustness of our method, by assuming that one among the four markers have slipped (i.e. perform a displacement which is not coherent with the other three). We have calibrated the M-estimator to saturate errors higher than 4 voxels. Doing so, is only possible if we have already a relevant initial estimation of the positioning, which is our case here.

As reported in Fig. 14, the behaviour of the estimator is “robust” in the sense that it “saturates” the error around 1 voxel, and limit the “influence”

Marker slip (in voxels)	0	0.1	0.2	0.5	1	2	4	5
Localisation precision (in voxels)	0.3	0.3	0.5	0.4	1.1	1.4	1.3	1.2

Figure 14: Precision in voxel for the localisation, given different slip of 1 marker among 4, for pure translations.

of the outlier (a linear estimator would have introduced an additional error of about 25 % whereas we obtain a bit less here). However, no miracle here, and methods which *compare the residual error using either all markers or only a subset of them* to decide whether they better eliminate a marker or not should be investigated when considering human experimentations.

Finally we have investigated the capability of the method to estimate, not only translations or exact rotations, but also other kind of displacements.

On one hand, we have evaluated when it was numerically a good deal to use 1st or 2nd order approximations of the rotation. This is qualitatively reported in Fig. 15. Here we verify that, in this paradigm:

- rotations smaller than 1 degree are negligible and are not detected,
- rotations below 5/10 degrees are easily estimated using a 1st order model,
- 2nd order approximation of the rotation is more or less not used.

Displacement	Pure translation $\theta = 0$ deg	Tiny rotation $\theta = 1$ deg	Small rotation $\theta = 5$ deg	True rotation $\theta = 10$ deg
Model chosen	Pure translation	Pure translation	First order rotation	Rigid displacement

Figure 15: Behaviour of the model selection mechanism.

This suggests in fact to avoid using a very complex hierarchy of models in this case, but only either the subset we have proposed, or even a smaller one. For instance (sorry ! but it was to be verified ..), 2nd order approximation of rotations matrices may be forgotten.

From a computational point of view, this allows to deal with a few pre-determined models, contrary to what was proposed in [18] for a more complex study.

On the other hand, we have evaluated how affine deformations could be corrected by our method. From a “theoretical” point of view (i.e. using a lot of big markers !) we are always able to correct for scale or skew deformations and it was “fun” to numerically verify that **but** : if it is realistic to consider *only 4 markers of 3/4 voxels size* the conclusions are rather pessimistic :

- only a global affine transformation may be estimated
- models considering scale deformations versus skew deformations are not differentiated
- local affine deformations is not captured by this approach.

We however have verified a very nice property: if we assume that scales are known whereas skew deformations occurs, i.e. in (2), we only write

$$\mathbf{S}(\mathbf{s}) = \begin{bmatrix} 1 & s_2 & s_1 \\ s_2 & 1 & s_0 \\ s_1 & s_0 & 1 \end{bmatrix}$$

these three parameters can easily be recovered in addition to the rotation and translation. In our numerical simulations we have obtained a qualitative precision of about 5 / 10 %.

This may be a track to use these markers to not only correct for motion displacements but also pre-estimate inter-modalities deformations.

But, at the present stage, since we do not have “real data” available but only simulations, we have no chance to really study these aspects.

6 Conclusion

We have experimented how suitable a rather “sophisticated” method of localisation of known small markers in a data volume may help to correct for rigid displacements and some deformations.

The present method had three goals :

Demonstrate that it is worth using such markers. We did it: contrary to other methods based on data comparisons, we have here a true, ground based, way of correcting these displacements and a very precise evaluation of the method precision: say, 0.3 voxel.

Experiment which kind of markers should be used. This is done: we have shown that using 4 markers (more is better for deformations estima-

tion but not for motion correction) seems ok, while we must recommend to have them as big as possible.

We also have seen that *cubic markers* would simplify and improve the numerical estimation process, which has to be discussed with neuroscientists.

Evaluate how sophisticated such a mechanism could be. Not that much: if rotation and translation is easily and accurately estimated, only *global deformations, function of a very small (say 2/3/4) number of parameters* could have been evaluated. However, our methodology allows to use *any non-linear transformation*, e.g. quadratic, inhomogeneous, only in the slices planes or orthogonal to them, etc...

A perspective of this work is thus to experiment, on real data, several non-linear transformations, proposed by MRI physicians, corrected by the “non-rigid residual error” of our localisation process.

As a consequence, this study will be pursued in the scope of a cooperative, inter-disciplinary project, as quoted in the acknowledgements.



Figure 16: A view of the interactive software used for the experimentations.

Thanks to suitable software developments (see Fig.16) the small modules developed here will be easily experimented, modified and upgraded within the scope of this project.

References

- [1] Y. Bar-Shalom and T. Fortmann. *Tracking and Data Association*. Academic, New York, 1988.
- [2] H. Ben-Ameur, G. Chavent, and J. Jaffré. Raffinement et de déraffinement de paramétrisation pour l'estimation de transmissivité hydraulique. RR 3623, INRIA, 1999.
- [3] S. S. Blackman. *Multiple-Target Tracking with Radar Application*. Artech House, Norwood, MA, 1986.
- [4] M. J. Boer. *Magnetic Resonance Imaging*. Springer, 1996.
- [5] T. Broida, S. Chandrashekhar, and R. Chellappa. Recursive 3-D motion estimation from a monocular image sequence. *IEEE TransactionsAES*, 26(4):639–656, July 1990.
- [6] O. Faugeras and M. Hébert. The representation, recognition, and locating of 3d shapes from range data. *The International Journal of Robotics Research*, 5:27–52, 1986.
- [7] K. Friston, C. D. Frith, R. S. J. Frackowiak, and R. Turner. Characterizing dynamic brain responses with fmri: a multivariate approach. *NeuroImage*, 2:166–172, 1995.
- [8] K. J. Friston and C. Büchel. A nonlinear analysis of functional brain architectures. *NeuroImage*, 7, 1998.
- [9] F. Gaspard and T. Viéville. Non linear minimization and visual localization of a plane. In *The 6th International Conference on Information Systems, Analysis and Synthesis*, volume VIII, pages 366–371, 2000.
- [10] G. Golub and C. van Loan. *Matrix Computations*. The John Hopkins University Press, 1989.
- [11] G. Hermosillo. Recalage par maximization de l'information mutuelle. Master's thesis, Université de Nice, Sophia, Sept. 1998.
- [12] G. Hochschild. *La Structure des Groupes de Lie*. Dunod, Paris, 1968.
- [13] P. Huber. *Robust Statistics*. John Wiley & Sons, New York, 1981.
- [14] V. S. Hwang. Tracking feature points in time-varying images using an opportunistic approach. *Pattern Recog.*, 22(3):247–256, 1989.
- [15] M. Kendall. *The Advanced Theory of Statistics*. C. Griffin and Co., 1960. 4th Edition.
- [16] S. Kullback. *Information Theory and Statistics*. Wiley, New-York, 1959.
- [17] D. Le Bihan. Methods and applications of diffusion mri. In I. Young, editor, *Magnetic Resonance Imaging and Spectroscopy in Medicine and Biology*. John Wiley and Sons, 2000.
- [18] D. Lingrand. *Analyse Adaptative du Mouvement dans des Séquences Monoculaires non Calibrées*. PhD thesis, Université de Nice - Sophia Antipolis, INRIA, Sophia Antipolis, France, July 1999.
- [19] P. Meer, D. Mintz, A. Rosenfeld, and D. Kim. Robust regression methods for computer vision: A review. *The International Journal of Computer Vision*, 6(1):59–70, 1991.

- [20] A. Morgan. *Solving polynomial systems using continuation for engineering and scientific problems*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
- [21] J. L. Mundy and A. Zisserman, editors. *Geometric Invariance in Computer Vision*. MIT Press, 1992.
- [22] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.
- [23] L. Robert. Camera calibration without feature extraction. *Computer Vision, Graphics, and Image Processing*, 63(2):314–325, Mar. 1995. also INRIA Technical Report 2204.
- [24] A. Roche, G. Malandain, X. Pennec, and N. Ayache. Multimodal image registration by maximization of the correlation ratio. Technical Report 3378, INRIA, Aug. 1998.
- [25] O. Rodrigues. Des lois géométriques qui régissent les déplacements d’un système solide dans l’espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire. *Journal de Mathématiques Pures et Appliquées*, 5, 1840. pp. 380–440.
- [26] T. Viéville, J. O. Eklund, K. Pahlavan, and T. Uhlin. An example of artificial oculomotor behavior. In T. Henderson, editor, *Seventh IEEE Symposium on Intelligent Control, Glasgow*, pages 348–353. IEEE Computer Society Press, 1992.
- [27] T. Viéville and D. Lingrand. Using specific displacements to analyze motion without calibration. *The International Journal of Computer Vision*, 31(1):5–29, 1999.
- [28] T. Vieville, D. Lingrand, and F. Gaspard. Implementing a multi-model estimation method. *The International Journal of Computer Vision*, 2001. in press.
- [29] T. Viéville, C. Zeller, and L. Robert. Using collineations to compute motion and structure in an uncalibrated image sequence. *The International Journal of Computer Vision*, 20(3):213–242, 1996.
- [30] C. Wampler, A. Morgan, and A. Sommese. Numerical continuation methods for solving polynomial systems arising in kinematics. Technical Report GMR-6372, General Motors Research Labs, Aug. 1988.
- [31] C. Westbrook and C. Kaut. *MRI in practice*. Blackwell scientific publications, 1993.
- [32] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *The International Journal of Computer Vision*, 13(2):119–152, 1994. also Research Report No.1658, INRIA Sophia-Antipolis, 1992.
- [33] Z. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and Vision Computing Journal*, 15(1):59–76, 1997.

Acknowledgements:

Pr. G. Orban, from Laboratorium Voor Neuro- En Psychofysiologie, Leuven and **Pr. O.D. Faugeras** from INRIA are gratefully acknowledged for powerful ideas which are at the origin of this work.

Dr. François Gaspard, alias *le Barbare*, is vachement acknowledged for his contribution to the “physical” parameter minimisation method, used in this paper.

M. Robert Fournier, is deeply acknowledged for his contribution to the software developments, which have allowed to experiment the algorithms given in this paper.

This work has been financed and realized within the scope of the ESPRIT MAPAWAMO project (this preliminary study is mandatory for the design of the project experimental paradigm).

A Using conic markers.

Here, we consider conic markers of maximal radius R and height H , as illustrated in Fig. 17.A.

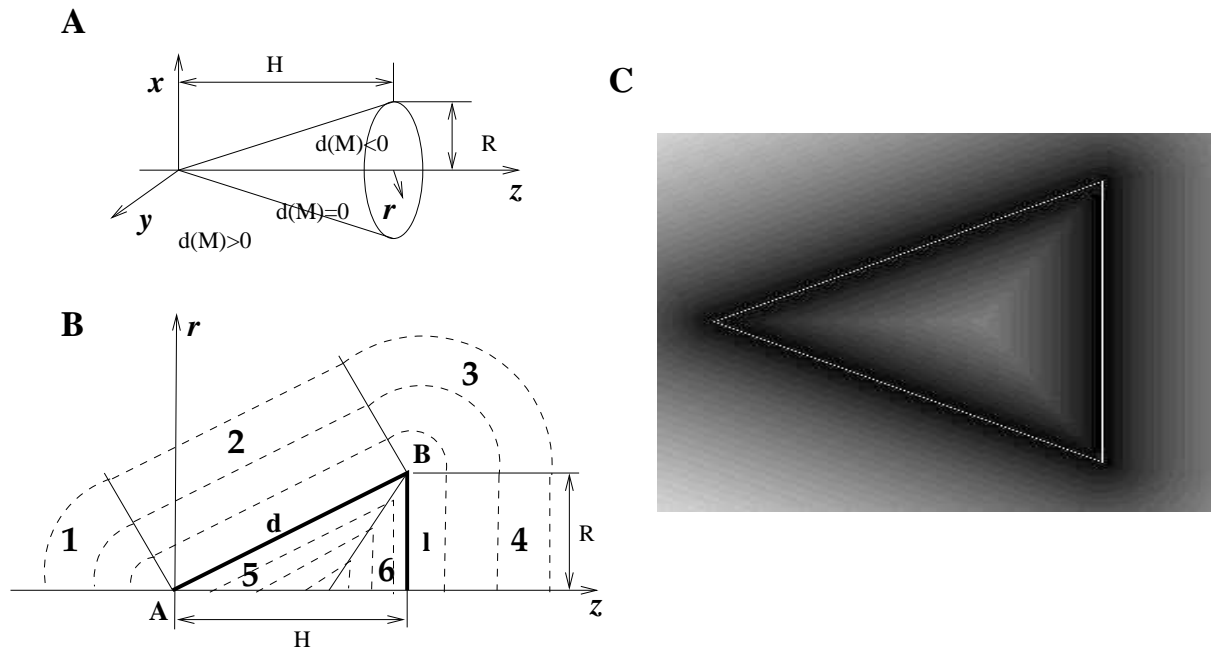


Figure 17: Computing the Euclidean distance to the conic marker surface, see text.

In order to compute the $\mathcal{R}^3 \rightarrow \mathcal{R}$ signed Euclidean distance to the conic marker surface, we consider for a given point (x, y, z) two of its cylindrical coordinates $(r = \sqrt{x^2 + y^2} \geq 0, z)$, as represented in Fig. 17.B. The conic surface now reduces to two line-segments \mathbf{d} and \mathbf{l} in shown half-plane.

Then, analysing the orientation of the point with respect to the points **A** and **B**, we obtain six areas shown in Fig. 17.B. Finding the distance to the conic surface reduces to the calculation of the distance to either the two points **A** and **B** or the two lines **d** and **l**, depending on the area.

This leads to the function given in Fig. 18 the result being shown in Fig. 17.C.

$$\begin{aligned}
 d(\mathbf{M} = (x, y, z)) = & \\
 \theta = \widehat{\vec{\mathbf{A}\mathbf{z}}, \vec{\mathbf{A}\mathbf{B}}} = & \arctan(R/H); \\
 r = & \sqrt{x^2 + y^2}; \\
 \text{if } r \tan(\theta) + z < 0 & \\
 \text{then } d(\mathbf{M}, \mathbf{A}) = & \sqrt{r^2 + z^2} \\
 \text{else} & \\
 \phi = \widehat{\vec{\mathbf{A}\mathbf{z}}, \vec{\mathbf{M}\mathbf{B}}} = & -\arctan(r - R, H - z); \\
 \text{if } \phi < \theta - \Pi/2 & \\
 \text{then } d(\mathbf{M}, \mathbf{B}) = & \sqrt{(r - R)^2 + (z - H)^2} \\
 \text{else if } \phi > \theta/2 + \Pi/4 & \\
 \text{then } d(\mathbf{M}, \mathbf{l}) = & z - H \\
 \text{else } d(\mathbf{M}, \mathbf{d}) = & (Hr - Rz)/\sqrt{R^2 + H^2}
 \end{aligned}$$

Figure 18: Computing the Euclidean distance to the conic marker surface, see text.

A step further, we have to compute the first and second order momenta of the conic volume and easily obtain :

$$\bar{v}_1 = \begin{pmatrix} 0 \\ 0 \\ 3/4 H \end{pmatrix} \text{ and } \bar{\Upsilon}_2 = \frac{15}{100} \begin{pmatrix} R^2 & 0 & 0 \\ 0 & R^2 & 0 \\ 0 & 0 & H^2/4 \end{pmatrix}$$

considering the frame of reference of Fig. 17.A.

These quantities are computed by integration, over the conic marker *volume*, of a function $f(x, y, z)$ defined at a point $M = (x, y, z)$:

$$I_f = \int_0^H \int_0^{\frac{Rz}{H}} \int_0^{\frac{\pi}{2}} f(r \cos(\theta), r \sin(\theta), z) r d\theta dr dz$$

Similarly, by integration over the conic marker *surface*, of a function $g(x, y, z; n_x, n_y, n_z)$ defined at a point $M = (x, y, z)$ of the surface, $\mathbf{n} = (n_x, n_y, n_z)$ being the normal at this point:

$$I_g = \int_0^H \int_0^{\frac{\pi}{2}} g(r \cos(\theta), r \sin(\theta), z; \cos(A) \cos(\theta), \cos(A) \sin(\theta), -\sin(A)) r d\theta dz + \int_0^R \int_0^{\frac{\pi}{2}} g(r \cos(\theta), r \sin(\theta), H; 0, 0, 1) r d\theta dr$$

with $\tan(A) = \frac{R}{H}$, from (12) we easily calculate, the information matrix:

$$\Lambda_q^{-1} = \frac{\|\bar{\mathbf{g}}\|^2}{V_I} \begin{bmatrix} k_1 & 0 & 0 & 0 & k_2 & 0 & 0 & 0 & 0 & k_3 & 0 & 0 \\ 0 & k_1 & 0 & -k_2 & 0 & 0 & 0 & 0 & 0 & 0 & k_3 & 0 \\ 0 & 0 & k_4 & 0 & 0 & 0 & k_5 & k_6 & 0 & 0 & 0 & 0 \\ 0 & -k_2 & 0 & k_7 & 0 & 0 & 0 & 0 & 0 & 0 & k_8 & 0 \\ k_2 & 0 & 0 & 0 & k_7 & 0 & 0 & 0 & 0 & -k_8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_5 & 0 & 0 & 0 & k_9 & -k_9 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_6 & 0 & 0 & 0 & -k_9 & k_{10} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{11} & 0 & 0 & 0 \\ k_3 & 0 & 0 & 0 & -k_8 & 0 & 0 & 0 & 0 & k_{12} & 0 & 0 \\ 0 & k_3 & 0 & k_8 & 0 & 0 & 0 & 0 & 0 & 0 & k_{12} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{11} \end{bmatrix} \quad (23)$$

with

$$\left\{ \begin{array}{l} k_1 = 1/2 \frac{\pi R H^3}{R^2 + H^2} \\ k_2 = 1/3 \pi R H^2 \\ k_3 = -1/3 \frac{\pi R H^2 (-H+R)(H+R)}{R^2 + H^2} \\ k_4 = \frac{R^2 \pi (R H + R^2 + H^2)}{R^2 + H^2} \\ k_5 = -\pi R^2 H \\ k_6 = 1/3 \frac{\pi R^2 H (4 R H + 3 R^2 + 3 H^2)}{R^2 + H^2} \\ k_7 = 1/4 (R^3 + R^2 H + H^3) R \pi \\ k_8 = 1/4 (R^3 + R^2 H - H^3) R \pi \\ k_9 = \pi R^2 H^2 \\ k_{10} = \frac{\pi R^2 H^2 (H+R)^2}{R^2 + H^2} \\ k_{11} = 1/4 \frac{R^3 H^3 \pi}{R^2 + H^2} \\ k_{12} = 1/4 \frac{\pi R (H^5 - 2 R^2 H^3 + R^4 H + R^5 + R^3 H^2)}{R^2 + H^2} \end{array} \right.$$

In particular, the rank of this information matrix is 11 and ranks of related information matrices for sphere, cylinder, cube, considering the different displacements, strictly correspond to what has been given in Fig. 5.

B Defining a M-robust semi-norm.

In this paper we have to design criteria of the form:

$$\mathcal{L}(\mathbf{q}) = \sum_k \rho(v_k(\mathbf{q}))$$

where $v_k(\mathbf{q})$ is a “measurement error” we want to minimise attempting to combine *robustness* with respect to outliers and *optimality* with respect to uncertainty.

Let us discuss how to choose such a $\rho()$ function.

In practice (e.g. [13], see [33] for a recent review in the field) one class of such methods (*M*-estimation) consist of using a function $\rho()$:

- (i) close to the square function for small errors, since this is optimal with respect to Gaussian additive noise,
- (ii) close to the absolute value function for higher errors, in order to derive an estimation with a small bias in the presence of outliers,
- (iii) with a saturation of large value, at, say, twice the level of (in)distinguishable values, in order to cancel the influence of aberrant data with a large error.

It appears [33] that the precise profile of the function $\rho()$ is not relevant since it depends upon the (unknown) statistics of the errors. Here, we have to choose a profile which eliminates influence of outliers and have adapted the Welsh operator as follows.

We consider a strictly increasing, symmetric, fully differentiable function, with a unique minimum, $\rho(\cdot)$ having the following¹¹ properties:

$$\begin{aligned} \rho(0) &= 0 & \rho'(0) &= 0 & \rho''(0) &= 2 & \rho'''(0) &= 0 & \rho^{(4)}(0) &= 0 \\ \rho(u) &\simeq \frac{\kappa}{2} & \rho'(u) &\simeq u & \rho''(u) &= 0 & \rho'''(u) &= 0 & \rho^{(4)}(u) &\simeq 0 \\ \rho(\infty) &= \kappa & \rho^{(n)}(\infty) &= 0 & \forall n > 0 & & & & & \end{aligned}$$

for which, with $\kappa = 2 u^2$:

$$\rho(\epsilon) = \kappa \left(1 - \left(1 + \frac{\epsilon^2}{\kappa} \right) e^{-2 \frac{\epsilon^2}{\kappa}} \right) = \epsilon^2 + O(\epsilon^6)$$

is a suitable candidate with $\rho(u) = 0.48 \kappa$ and $\rho'(u) = 1.47 u$, as shown in figure 19.

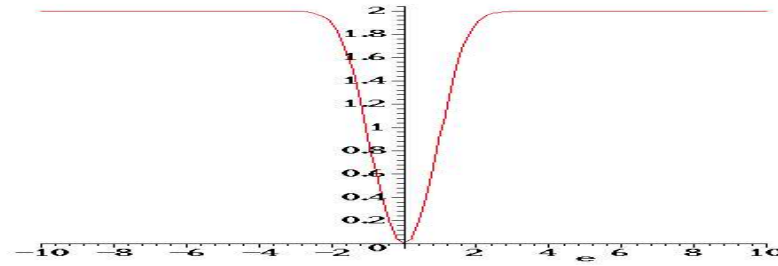


Figure 19: The M-estimator profile used in the estimation process, for $\kappa = 2$.

With this profile, if $|\epsilon| > \epsilon_{max}$ with $\epsilon_{max} \simeq 4 u$ the function is 99.9% saturated and the term has no influence anymore. As a consequence, this automatically eliminates the influence of “gross” errors.

¹¹These properties state that :

- (i) $\rho(\cdot)$ is close to $\epsilon \rightarrow \epsilon^2$ around zero,
 - (ii) $\rho(\cdot)$ is not far from $\epsilon \rightarrow |\epsilon|$ for intermediate errors (half of the saturation),
 - (iii) $\rho(\cdot)$ saturates for larger errors,
- as required by the specifications.

In fact, more stronger constraints can easily be stated and the corresponding function can always be numerically computed and hence tabulated to be used in the minimisation process ... but it appears that the practical gain is negligible, whereas the proposed function is a nice compromise.

This function is convex for $|\epsilon| < u$ as required for the minimisation process. Its *asymptotic efficiency*¹² on the standard distribution for $|\epsilon| < u$ is 97 % thus better than usual M -estimators [33] since it is closed to a least-square criterion for small values.

Here, for small residuals the criterion may be treated as a Ξ^2 criterion (i.e. a Mahalanobis distance) providing that errors on the residuals are additive and follow a standard Gaussian distribution.

¹²This efficiency is defined by the ratio: $\lim_{n \rightarrow \infty} \frac{\text{Var}(s_n)/E(s_n)^2}{\text{Var}(r_n)/E(r_n)^2}$ where s_n is an estimation obtained from a Gaussian optimal operator and r_n obtained from the M -estimator.



Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot St Martin (France)

Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399