Three-Dimensional Fuzzy Kernel Regression Framework for Registration of Medical Volume Data

Roberto Gallea, Edoardo Ardizzone, Roberto Pirrone, Orazio Gambino
Viale delle Scienze, Ed. 6, III Piano, 90128 Palermo, Tel. +399123862616

Abstract

In this work a general framework for non-rigid 3D medical image registration is presented. It relies on two pattern recognition techniques: kernel regression and fuzzy c-means clustering. The paper provides theoretic explanation, details the framework, and illustrates its application to implement three registration algorithms for CT/MR volumes as well as single 2D slices. The first two algorithms are landmark-based approaches, while the third one is an area-based technique. The last approach is based on iterative hierarchical volume subdivision, and maximization of mutual information. Moreover, a high performance Nvidia CUDA based implementation of the algorithm is presented.

The framework and its applications were evaluated with a number of tests, which show that the proposed approaches achieve valuable results when compared with state-of-the-art techniques. Additional assessment was taken by expert radiologists, providing performance feedback from the final user perspective.

Keywords: non-rigid registration, fuzzy regression, mutual information, interpolation, GPU computing

1. Introduction

One of the most widely accepted methods of gathering knowledge about tissues, organs and cells, is to integrate information integration coming from volumes/images of such objects that have been acquired with different modalities, different acquisition techniques, and at different times. Volume registration is a mandatory task to achieve information fusion. Registration lets the two volumes to be transformed geometrically so that the best possible...
spatial correspondence with respect to each other is obtained. Image and volume registration techniques span a broad class of methods and taxonomies according to either the features used to perform registration or the nature of transformation. Several surveys on the subject are present in literature, [1, 2, 3, 4, 5] and this research field is very active, as it is reported in [6].

With regard to the features, registration methods can be landmark-based or area-based. Landmark-based approaches rely on the information provided by some corresponding features in the two images, such as points, lines, regions, etc. Area-based techniques use the whole image content to estimate the registration transformation by optimizing some similarity metric. Although several similarity metrics have been proposed in literature, Mutual Information (MI) and its normalized version (Normalized Mutual Information - NMI) has proven to be one of the most effective measures, especially for multi-modality registration tasks [7, 8, 9, 10], since it does not assume any functional relationship between the intensity values of the images, taking into account only their statistical correspondence.

With regard to the nature of transformation, many models exist in literature. The simplest ones use global or local mapping by means of rigid, affine, and projective transformations. Other approaches are able to deal with local deformations and use radial basis functions such as Thin-plate spline [11] or Wendland’s functions [12, 13]. A more complex approach is to use parameter-free deformation functions, by considering the volume as a tensile material [14] or a viscous fluid [15] that is deformed by external and internal forces subject to constraints. In this approach, registration is achieved by the iterative minimization of an energy functional.

Another approach called block matching [16], finds local correspondences and then derives the global rigid transformation that best explains them. ANIMAL [17], realizes the registration using a two step registration (a linear and a non-linear part) relying on geometrically invariant spatial features. Polyaffine framework [18] parameterizes deformations with a finite number of rigid or affine components. Lastly, in MIRT [19] a gradient descent method is used hierarchically to iteratively determine optimal B-spline parameters for the transformation.

Using a global method is a practicable choice only when using simple transformation models, where just few parameters are required. When using curved deformations the number of parameters is large, and a direct optimization is not possible due to large dimensionality of the search space and the presence
of many local optima. A possible solution is to decompose the image domain and operate many local sub-image registrations using simple models. The final global transformation can be recovered by composing local ones, thus obtaining a unique continuous and smooth complex deformation [20, 21]. Such idea, and a variety of methods to recover a mapping function using points correspondences has been extensively investigated in [22] and [23]. Recently we proposed some 2d registration systems leveraging onto these concepts [24, 25, 26]. However, such image-based approaches were lacking a formal theoretical background. The present work, deepens the formal aspects. Additionally, the method proposed in this paper unifies the three approaches a unique 3D registration framework that relies on using kernel regression and fuzzy c-means clustering for recovering the required volume transformation parameters. We called such framework 3D fuzzy kernel regression. Three applications are presented as different instances of the framework where increasing complexity transformations are addressed. The first two are a simple and an improved landmark based technique (SLB and ILB) while the third one is an automatic area based approach (AAB) that addresses several hot problems in the field of registration: it does not require correspondences, achieves a per voxel transformation, and is inherently parallel. The presented techniques are compared to each other to prove the generality of the framework, while the AAB algorithm is compared with MIRT, and an overall performance review is provided by a team of radiologists.

The paper is arranged as follows: in section 2 theoretical background related to kernel regression and fuzzy c-means is reported. The three different framework implementations are illustrated in section 3, while section 4 reports the details about the implementation of the framework on the Compute Unified Device Architecture (CUDA) for increasing performance using NVidia GPUs to make some of the calculations. In section 5 the proposed registration methods are tested to evaluate their performances from both qualitative and quantitative perspective. Finally, in section 6 final considerations and future works are explained.

2. Theoretical framework

The proposed registration framework relies on two main theoretical concepts: Kernel Regression and Fuzzy c-means. For this reason, we will provide an explanation of such issues, and how they are used together for the purpose of volume data registration before illustrating the applications of the
proposed framework.

2.1. Kernel regression

Consider a target volume $T$, an input volume $I$ and a set of known displacements $t_n$ for some given pairs of corresponding points $c_{Tn}$ and $c_{In}$. In what follows, boldface notation will indicate 3D vectors and/or points. The registration problem can be stated as recovering the values for reconstructing the whole deformation function $g(x)$ which brings $I$ to the best spatial correspondence with $T$. In our work this was accomplished using kernel regression. Kernel regression is a *memory-based* pattern recognition method, i.e. it uses data points both in the training and in the prediction phase. It consists in predicting the function value for given input points by means of linear combinations of a *kernel function* evaluated at the training data points. Kernels depending only on the magnitude of the distance between their argument and the training points, are known as *homogeneous* kernels or *radial basis functions*. In our work we used the derivation of kernel regression from the scheme known as the Nadaraya-Watson model [27]. Starting from the training set made by $N$ couples $(c_n, t_n), \ i = 1, \ldots, N$, the joint distribution $p(x, t)$ can be modeled using a Parzen density estimator:

$$p(x, t) = \frac{1}{N} \sum_{n=1}^{N} f(x - c_n, t - t_n) \quad (1)$$

where $f(x, t)$ is the component density function. There is an instance of $f(\cdot)$ centered in each sub-image. The regression function $y(x)$, corresponding to the conditional average of the target variable depending on the input, is given by

$$y(t) = E[t|x] = \int_{-\infty}^{+\infty} t p(t|x) \, dt = \frac{\int t p(t|x) \, dt}{\int p(t|x) \, dt} = \frac{\sum_{n} \int t f(x - c_n, t - t_n) \, dt}{\sum_{m} \int f(x - c_m, t - t_m) \, dt}. \quad (2)$$

Assuming that the component density functions have zero mean so that

$$\int_{-\infty}^{+\infty} f(x, t) t \, dt = 0 \quad (3)$$

for all values of $x$, we can operate a variable substitution, and we get

$$y(x) = \frac{\sum_{n} g(x - c_n) t_n}{\sum_{m} g(x - c_m) t_m} = \sum_{n} k(x, c_n) t_n, \quad (4)$$

4
where the kernel function \( k(x, c_n) \) is defined as

\[
k(x, c_n) = \frac{g(x - c_n)}{\sum_m g(x - c_m)}
\]  

(5)

and

\[
g(x) = \int_{-\infty}^{+\infty} f(x, t)dt.
\]  

(6)

This form is known as the Nadaraya-Watson model or kernel regression [27], [28]. In case of localized kernel functions, it has the property of weighting more the data points \( c_n \) close to \( x \) than the others. The kernel (5) satisfies the summation constraint

\[
\sum_{n=1}^{N} k(x, c_n) = 1.
\]  

(7)

2.2. Fuzzy c-means clustering

In order to use the kernel regression model we need to choose a suitable equivalent kernel which satisfies (7). Several functions can be chosen for this purpose, such as Gaussians, multiquadric, polyharmonic, Thin-plate splines [11], etc. In our framework we introduce fuzzy membership maps as equivalent kernels. Such functions are designed using fuzzy c-means (FCM) clustering algorithm [29]. Given a training set of feature vectors \( \{x_i, i = 1, \ldots, k\} \) that defines a feature space \( \Omega \), FCM finds analytically the position of the cluster centroid vectors \( \{c_j, j = 1, \ldots, m\} \) in such a space. This is accomplished by minimizing the following functional:

\[
J_s = \sum_{j=1}^{m} \sum_{i=1}^{k} (u_{ij})^s d(x_i, c_j)^2, \quad 1 \leq s < \infty,
\]  

(8)

where \( d(x_i, c_j) \) is a distance function between each observation vector \( x_i \) and the cluster centroid \( c_j \), \( m \) is the number of clusters, which should be chosen a priori, \( k \) is the number of observations, \( u_{ij} \) is the membership degree of the sample \( x_i \) belonging to the \( j \)-th cluster and \( s \geq 1 \) is a parameter which defines the amount of clustering fuzziness, i.e. the form of the membership function. For common tasks this value ranges generally in an interval around 2.
additional constraint is that the membership degrees should be positive and structured such that
\[ \sum_{j=1}^{m} u(x, c_j) = 1. \]  \hspace{1cm} (9)

The method proceeds as an iterative algorithm where, at each step, given the membership matrix \( U = [u_{ij}] \) of size \( k \) by \( m \), the new positions of the centroids are updated according to the following equation:

\[ c_j = \frac{\sum_{i=1}^{k} u_{ij} x_i}{\sum_{i=1}^{k} u_{ij}}. \]  \hspace{1cm} (10)

The new membership values \( u_{ij} \) are recovered as:

\[ u_{ij} = \frac{1}{\sum_{l=1}^{m} \left( \frac{d(x, c_l)}{d(x, c_j)} \right)^{r-1}}. \]  \hspace{1cm} (11)

2.3. 3D Fuzzy Kernel regression

Due to the summation constraint (9), fuzzy membership values can be used as equivalent kernels for the purpose of kernel regression. For our task, the feature space \( \Omega \) is represented by the volume spatial domain. Each cluster defines a spatial region subject to a local deformation. This brings several advantages: first, no clustering algorithm should be actually executed since the cluster centroids are already known, being them the centers of the defined sub-volumes (i.e. the FCM concepts are used just to extract membership functions). The membership maps can be evaluated at startup and used successively as a look-up table. In addition such evaluation is simple to compute since it is defined just by a combination of sums and products of distance measures.

The unified 3D Fuzzy Kernel Regression framework then becomes:

\[ y(x) = \sum_{n} u(x, c_n) t_n, \]  \hspace{1cm} (12)

where \( u(x, c_n) \) is the fuzzy membership map of each point \( x \) relative to the cluster centroid \( c_n \), and \( t_n \) is the assumed target variable. By defining the
prior information in terms of the sub-regions centroids $c_n$ and plugging specific target variables $t_n$ formulation is possible to instantiate different framework applications.

As a consequence of the whole process, each voxel will be subject to a motion vector whose direction and intensity are influenced by all of the local transformations recovered for each sub-volume. The closest sub-volumes will influence the vector at the maximum degree, while the furthest ones will provide progressively decreasing contributions. This results in a continuous and smooth 3D deformation field. The amount of smoothness is governed by the fuzziness value $s$, which is the unique tunable parameter of the whole process, and has been set in the range $[1.6, 2]$ in our applications. As can be seen from its formulation, the 3D Fuzzy Kernel Regression approach has several advantages, which are listed below:

- Low computational effort: neither iterative nor analytical minimization procedure is required, since membership maps are recovered from distance measures and algebraic operation only (note that clustering is NOT actually required).

- General concept: since the framework results from a general formulation, it can be applied in several ways, by just changing the prior information in terms of definition of the assumed target variables.

- Embarrassing parallelism: due to point-based nature of the framework, its parallel implementation is straightforward and can give significant performance improvements.

3. Framework application

The framework described above can be applied in several ways depending on the criteria used to define prior information in terms of both the centroids $c_n$ in sub-volumes and the target variables $t_n$, i.e. what type of local deformation is used.

In the following paragraphs three registration techniques are described. The first two implementations are feature-based approaches. Despite such methods are less time-consuming, they require landmark points to be either defined manually or detected automatically. The third application is a fully automatic area-based approach. It makes use of Normalized Mutual Information as the optimization function to align input and target volumes.
3.1. Simple landmarks based registration (SLB)

The first application of the 3D Fuzzy Kernel Regression framework arises naturally from its definition. It is really simple and straightforward, and it is effective notwithstanding its simplicity, although it is meant to demonstrate the actual use of the registration scheme. As a consequence it could be used even for real registration tasks. In this scheme, sub-volumes centers represent the cluster centroids $c_n$ for the FCM and are chosen as a set of landmark points. Landmarks can be either picked manually or extracted automatically with suitable feature detectors such as [30, 31]. Target variables $t_n$ are represented directly by the displacement vectors $d_n$ of the corresponding features. Plugging them to (12), the resulting system is defined by the following equation:

$$ y(x) = \sum_{n} u(x, c_n) d_n, $$

(13)

After computing membership maps $U$, eq. (4) gives a displacement vector at each voxel that is computed as the weighted contribute of each sub-volume center displacement. Such a contribute is high for points that are close to landmarks, and gets smaller as relative distances between the considered input point and the landmarks increase.

An example of the $x$, $y$ and $z$ displacement maps recovered for one of the test volumes is reported in figure 1.

![Displacement maps](image)

Figure 1: Displacements surfaces recovered using simple landmark based fuzzy kernel regression registration: the two maps represent $x$ (a), $y$ (b), and $z$ (c) direction displacements across an entire volume.

3.2. Improved landmarks based registration (ILB)

Although the simple method described previously is effective and can be useful for simple registration tasks, it is not suitable for many applications,
between two different regions of the volume are continuous but not smooth since the resulting displacement surface will present crisp edges. Transitions registered volume. Anyway, this direct composition is not sufficient per se, while composing all the transformations allows the full reconstruction of the volume. Each transformation is recovered from a tetrahedrons pair correspondence, of vertices. Given the set of landmark points and their correspondences, the tessellation produces a tetrahedrons set along with their relative vertices correspondences. The second step consists in recovering the local transformations of each region by finding the affine transformations, which map each tetrahedron in the input volume \( I \) to its respective counterpart in the target volume \( T \). The affine matrices are fully determined by twelve parameters. Writing down the transformation equations (14) for four points yields a linear system of twelve equations to recover such parameters. Such parameters are determined writing and solving the linear system associated to the four vertexes of each tetrahedron.

\[
\begin{bmatrix}
  x \\
  y \\
  z \\
  1
\end{bmatrix} = \begin{bmatrix}
  a & b & c & d \\
  e & f & g & h \\
  i & l & m & n \\
  0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  x_0 \\
  y_0 \\
  z_0 \\
  1
\end{bmatrix} = \begin{bmatrix}
  ax_0 + by_0 + cz_0 + d \\
  ex_0 + fy_0 + gz_0 + h \\
  ix_0 + ly_0 + mz_0 + n
\end{bmatrix} \Rightarrow
\]

(14)

Each transformation is recovered from a tetrahedrons pair correspondence, while composing all the transformations allows the full reconstruction of the registered volume. Anyway, this direct composition is not sufficient per se, since the resulting displacement surface will present crisp edges. Transitions between two different regions of the volume are continuous but not smooth
due to the adjacency of tetrahedrons faces and edges. This can lead to severe artifacts in the registered volume, especially in points outside the convex hull determined by the tessellation, where no transformation information exists. Such problem is shown for the 2D case in fig. 2c-d and fig. 3a-b. The example makes use of single pairs of 2D slices without loss of generality: pixels can be used in place of voxels simply dropping the z coordinate in the framework equations. Figures depict a registration example along with the plot of the recovered displacement surfaces. 3D Fuzzy Kernel Regression can be used to overcome the non smoothness drawback. Since the method requires the use of sub-volume centers \( c_n \) and target variables \( t_n \), we chose purposely the tetrahedrons centers of mass as clusters centroids \( c_n \), while we used the values obtained from affine transformations as target variables \( t_n \). In this way, after recovering the membership matrix \( U \) acting as kernel function for the regression, the final displacement for each input point is given by the weighted sum of the displacements resulting by all of the affine matrices. In conclusion, whole image information is taken into account. Kernel regression equation (12), i.e. the registration function, can then be expressed as follows:

\[
y(x) = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \sum_n \left( u(x, c_n) \begin{bmatrix} a_n & b_n & c_n & d_n \\ e_n & f_n & g_n & h_n \\ i_n & l_n & m_n & n_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \right).
\] (15)

In this way there are no more displacements values that change sharply when crossing tetrahedrons faces, and smooth variations are obtained. In fig. 2e-f and fig. 3c-d a ILB registration example and the corresponding deformation surfaces are shown, and they are compared with the non smooth registration reported in the line above for each figure. Readers should note the absence of sharp edges in the surface plots and the presence of displacement values also for points outside of the convex hull defined by the landmark points.

### 3.3. Automatic area-based registration (AAB)

Our last implementation is an area-based method, where no landmark points need to be selected since correspondences are determined during the registration process itself. In this case the problem of aligning the input volume \( I \) and the target volume \( T \) can be represented as the problem of finding the optimal spatial transformation \( F^* \) able to match \( I \) and \( T \). Such parameters are found by maximizing a similarity function \( E \):

\[
F^* = \arg\max_{F \in \Gamma} \{ E(F) \} = \arg\max_{F \in \Gamma} \left( E_{\text{sim}}(T, I \circ F) - E_{\text{reg}}(F) \right)
\] (16)
Figure 2: Example of improved landmark based 2D registration. The 2D case has been used without loss of generality to improve the figures readability. Input image (a), target image (b), deformed grid (c) and registered image (d) using simple affine transformations composition, deformed grid (e) and registered image (f) using fuzzy kernel regression affine transformations composition. In this example 18 landmark points where used.

where $\Gamma$ is the space of all the admissible transformations. The term $I \circ F$ represents the transformation of $I$ subject to $F$. $E$ is divided into two parts: the data similarity term $E_{\text{sim}}$ and an optional regularization term $E_{\text{reg}}$, which is used to penalize undesired transformations. $E_{\text{sim}}$ can be designed using several functionals: in our work we adopted Normalized Mutual Information (NMI). Therefore the similarity term becomes as follows:

$$E_{\text{sim}} (T, I \circ F) = Y (T, I \circ F) = \frac{H (T) + H (I \circ F)}{H (T, I \circ F)}$$

where $H (T)$ and $H (I \circ F)$ are the entropies of the target volume and of the transformed input volume respectively, while $H (T, I \circ F)$ is their joint entropy. Since in our method we use piece-wise registration, the minimization of the cost function is performed separately in each sub-region in order
Figure 3: Displacements surfaces recovered using simple affine transformations composition (first row) and fuzzy kernel regression affine transformations composition (second row). $x$ (first column) and $y$ (second column) direction displacements are shown for the same slices depicted in fig. 2.

to recover local registration transformations. The composition of such deformations is then operated by means of fuzzy kernel regression.

The regularization term $E_{reg}$ is intrinsically taken into account by the framework itself. In particular, the regularization is operated by the fuzziness exponent $s$, which is responsible for the plasticity/elasticity of the local transformations. When $s \approx 1$ the transformation is almost plastic, clustering is almost crisp, membership maps are box-shaped and regions are registered with affine transformations that are not blended. When $s \to \infty$, membership maps approaches impulses, transformations are “liquid” and just the centroids of the clusters are moved, leaving the other voxels unchanged. Using one of the in between values (generally around 1.4 and 1.8) allows smooth blending of the local transformations and produces a regularized global transformation. See Fig. 4 for reference, where five landmark points are used at the corners and at the center of the image, and the center point is moved in the South-East direction.

The complete registration algorithm is realized by several steps, which are summarized in the block diagram reported in fig. 5. Since mutual information is sensitive to noise, a preprocessing step aimed to noise reduction
Figure 4: Regularization effect of the parameter $s$. Its value vary from 1.1 (no regularization) to 2.9 (over regularized). Top row: deformed pattern. Bottom row: Resulting membership maps.

Figure 5: Block diagram for the area-based fuzzy kernel regression registration algorithm. The first block performs a binary volume pre-processing. The second one computes a coarse global affine registration used as starting condition for the elastic registration procedure, which runs three steps iteratively. First a hierarchical volume subdivision is operated, then the resulting sub-volumes are aligned locally, and a smooth composition of the registered volume is achieved by means of fuzzy kernel regression.

is applied to mitigate this weakness: a binary mask is used slice by slice to separate the content from the background, which is discarded. The second step consists in the application of a global affine transformation to the pre-processed data volume. Such a transformation will be used as the starting point for the successive elastic registration [33]. Such strategy reduces large misalignments and provides a speed up for the convergence of the successive steps. The core of the method is the elastic registration. This step is realized with the smooth composition of several local affine transformations. In particular, such transformations are evaluated hierarchically with a coarse to fine strategy i.e., the extent of the region subject to the effect of the computed transformations gets smaller and smaller throughout the evolution of the procedure. This allows aligning fine details too. For this purpose the input volume is subdivided into several sub-volumes. Then, for each sub-volume, the local optimum affine transformation aligning it to the target volume is computed maximizing the NMI measure. The recovered transformations are
then propagated throughout the entire volume using a composition based on the fuzzy kernel regression model. A huge amount of computation is required for the solution of the problem described above, so particular care should be taken for keeping the process affordable and efficient. To speed up convergence a multi-resolution pyramidal approach has been used. A three-level pyramid has been built from input and target volumes. At each level the volumes are subsampled with different factors, which are 8, 4 and 2. The deformation is evaluated iteratively at each level. The deepest is the level, the finer are the details that can be taken into account. At each step, the centers of the resulting regions act as the cluster centroids $c_n$. The fuzzy membership maps $U$ can be recovered starting from this arrangement. Note that centroids displacements need to be evaluated only once for each level of detail, while they are stored into a lookup table during the iterations at a certain resolution. Each local affine transformation matrix recovered from each sub-region alignment represents the target variable $t_n$. Once these values are known, local deformations can be composed as in the ILB approach using (15). As a consequence of the whole process, each voxel will be subject to a motion vector whose direction and intensity are influenced, with the proper extent, by all the local transformations recovered for each sub-volume. The closest regions will influence the vector at the maximum degree, while further ones will provide progressively minor contributions. The resulting deformation surface will be continuous and smooth. As for the other methods, the amount of smoothness is governed by the fuzziness tunable value $s$.

In fig. 6 the hierarchical approach used for 3D registration is shown, where the volume is partitioned into progressively smaller sub-volumes as level of detail increases. Sub-volumes are pair-wise registered by means of affine transformation matrices parametrized by NMI maximization. After all sub-volumes are registered, the resulting pieces are composed using fuzzy kernel regression to obtain a unique transformed volume.

3.4. Memory issues

This section reports some considerations holding for memory occupancy of the three proposed methods. Comparing the size of data structures, it results that in SLB $D \times M$ values need to be stored for landmark displacements, where $D$ is the dimensionality of a volume and $M$ the number of landmark points. Additional $M$ values are needed for the membership degrees of each point. Moreover, once every single voxel has been transformed, its membership degrees can be dropped so the total data structure is $M \times (D + 1)$ large.
Global
Local
Local

{Level 1
{Level 2
{Level n

Figure 6: Hierarchical approach to elastic volume registration.

ILB has a larger descriptor that is variable in size since it depends on the number of tetrahedrons in which the volume is subdivided. Such number is bounded to an order of $2^M$ anyway. Since each affine transformation is defined by $D \times (D+1)$ parameters and membership degrees require $2^M$ additional values (i.e. one for each tetrahedron), the whole registration function descriptor is in the order of $2^M \times [D \times (D+1) + 1]$. Similar considerations hold for AAB, except that regions are shaped as parallelepipeds and are not tetrahedrons. As a consequence, if $R$ is the number of regions used to decompose the volume, $R \times [D \times (D+1) + 1]$ values needs to be stored at each level of the hierarchy. As a result, the method has linear complexity w.r.t. the number of sub-regions $R$.

4. Parallelization on GPU clusters running CUDA

In order to improve execution performance, some operations have been parallelized using NVidia CUDA enabled devices.
CUDA (Compute Unified Device Architecture) is a hardware architecture introduced by Nvidia for parallel computing. It uses the concepts of kernels, which are function ran in parallel. In addition two hierarchies are at its base: thread hierarchy and memory hierarchy. A thread is a running instance of a kernel. Threads can be grouped into blocks, which can share some local data, blocks are grouped into a computational grid. Several memory level exist, which are basically: global memory, which is slow but very large and accessible by every thread; shared memory, which is faster but smaller than global one, and it’s accessible just by threads belonging to the same block and registers. Shared memories are very fast but they’re private to each single thread and reduced in size and number.

Such architecture can be exploited for general purpose programming by using its API, which is basically a C extensions. For further details on CUDA and its API refer to [34].

Basically, the following operations have been parallelized in the system, which is intended to be AAB for the rest of the dissertation:

- **Joint histogram generation**: it is used for (N)MI computation, and represents the co-occurrence matrix of each intensity level into the two volumes taken into consideration.

- **Fuzzy c-means algorithm**: it’s needed for speeding up the generation of fuzzy maps, which are used for kernel regression.

### 4.1. Joint histogram generation

Apparently, joint histogram generation is an extremely good candidate for parallelization; actually it is hard to be implemented efficiently. The operation consists in incrementing by one the histogram bin located at the coordinates defined by the intensity values of the floating and reference volume. Although this is an embarrassingly parallel operation in theory, a problem occurs. If the histogram updates are made in parallel, synchronization issues known as *race conditions* occur. Parallelizing a histogram with $B$ bins over $N$ threads is shown schematically in fig. 7. Updates to the histogram memory are data dependent, and they can results in race conditions and memory access conflicts. For this reason one of the main problem concerns the resolution of such conflicts. A typical solution for medium sized histograms, is to produce several sub-histograms with conflicts-free access, and then combine them into the final histogram. For a joint histogram of size $256 \times 256 \times 4$
Figure 7: Scheme for the parallel calculation of an histogram with $B$ bins distributed over $N$ threads. Update conflicts make necessary the synchronization of the threads to the device memory containing the histogram.

byte = 256 kB, this is possible. However, this size is larger than the actual shared memory size, so the histogram computation has to be segmented too. The proposed computational schema is shown in fig. 8.

Figure 8: Scheme for Joint Histogram computation over 3 thread blocks. Each block computes a segment of its own sub-histogram. At the end of each time step, the result is updated into the global memory to obtain the complete segment.

Each dataset is split into partitions, which are delegated to each thread block (three in the example). At each step, a thread block computes a sub-histogram for a single segment. Before proceeding with the next segment, the current one is updated to global memory. When using a device with
CUDA Compute Capability 2.0+ (i.e. a “Fermi” architecture GPU), such steps can be performed in parallel using streams. The last consideration is for global memory update. Since CUDA provides atomic operations starting from Compute Capability 1.1, not all of the GPUs allow using streams. Then, for old generations GPUs, the approach is to make each thread block updating each sub-histogram segment into different global memory location, and subsequently perform a reduction to the complete histogram with log(n) steps where n is the number of sub-histograms.

### 4.2. Fuzzy c-means algorithm

Fuzzy c-means clustering process consists essentially in the manipulation of the data matrices containing the data points and cluster centres. This can be efficiently accomplished using GPUs since there is low data dependency and high parallelization rates can be achieved. The computation is spread across several kernels, each one performing some processing. The six-pass procedure for FCM is shown in Figure 9.

Kernel 1 computes the (euclidean) distance matrix $D = [d(x_i, c_j)]$ from $X = [x_i]$ and $C = [c_j]$. Kernel 2 takes $D$ and computes the new membership values $M = [u_{ij}]$ according to equation 11. Membership values are raised directly to s-th power because they’re used in this way for the rest of the FCM procedure. Kernel 3 multiplies the membership values by the data points, creating the individual terms for the summation in the numerator of equation 10, which updates the centroids $c_j$. Kernels 4 and 5 operate respectively the two reductions aimed at computing the summation of the values in both the numerator and the denominator in equation 10. Finally, kernel 6 divides numerators by denominators. Reduction is an operation repeated over a series of elements to produce a final scalar value. Examples of reductions are the sum, min and max operators. A parallel reduction algorithms takes $\ln(N)$ number of passes, since at each time step is processed a fraction of the previous time step results. The first pass processes $N/2$ elements, the second $N/4$, and so forth. A graphical example of reduction is shown in Figure 10.

In Table 1 and Figure 11 is reported a plot showing the speedup of GPU versus CPU fuzzy c-means clustering. This example reports the results using 4096 data points, 64 clusters and varying the feature space dimensionality between 8 and 128. Tests were performed on a Nvidia Tesla C2070 GPU.

The outcomes point out that the performance increases linearly as the feature space dimensionality grows. Thus, the speed up becomes higher as
Figure 9: GPU algorithm for fuzzy clustering. \( X \) is the dataset, \( C \) are the cluster centers, \( M \) are the membership values and \( D \) the distance matrix. Kernel 1 computes the distance matrix, kernel 2 updates the membership values, kernel 3 computes the numerator for centers update equation, kernels 4 and 5 operate the reduction of the numerator and denominator of the centers update equation, and kernel 6 updates actually the centers values.

Figure 10: Parallel reduction scheme. At each iteration, elements are reduced to an half.

the feature space becomes larger. Finally, in Table 2 absolute performances for various large and very large clustering profiles are reported.
Table 1: Speedup measured as GPU/CPU ratio for fuzzy c-means clustering using 4096 data points, 64 cluster varying the feature space dimensionality.

<table>
<thead>
<tr>
<th>Dimensionality</th>
<th>GPU/CPU ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>29,852</td>
</tr>
<tr>
<td>16</td>
<td>44,216</td>
</tr>
<tr>
<td>24</td>
<td>50,101</td>
</tr>
<tr>
<td>32</td>
<td>42,600</td>
</tr>
<tr>
<td>40</td>
<td>49,984</td>
</tr>
<tr>
<td>48</td>
<td>56,525</td>
</tr>
<tr>
<td>56</td>
<td>63,742</td>
</tr>
<tr>
<td>64</td>
<td>69,978</td>
</tr>
<tr>
<td>72</td>
<td>73,847</td>
</tr>
<tr>
<td>80</td>
<td>75,266</td>
</tr>
<tr>
<td>88</td>
<td>78,215</td>
</tr>
<tr>
<td>96</td>
<td>85,488</td>
</tr>
<tr>
<td>104</td>
<td>80,946</td>
</tr>
<tr>
<td>112</td>
<td>78,585</td>
</tr>
<tr>
<td>120</td>
<td>92,352</td>
</tr>
<tr>
<td>128</td>
<td>104,445</td>
</tr>
</tbody>
</table>

Figure 11: Speedup measured as GPU/CPU ratio for fuzzy c-means clustering using 4096 data points, 64 cluster and varying the feature space dimensionality.

5. Tests and experimental results

The framework and all the proposed implementations have been extensively tested in order to evaluate their performance both quantitatively and qualitatively. First, experiments were conducted on the theoretic kernel regression framework, to determine its precision and applicability. Then, the
Table 2: Time in seconds for various profiles on the Nvidia Tesla C2070.

<table>
<thead>
<tr>
<th>Clustering profile</th>
<th>Time in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>C=4, DP=4096, F=4</td>
<td>0.005</td>
</tr>
<tr>
<td>C=4, DP=4096, F=128</td>
<td>0.009</td>
</tr>
<tr>
<td>C=64, DP=4096, F=4</td>
<td>0.021</td>
</tr>
<tr>
<td>C=64, DP=8192, F=4</td>
<td>0.029</td>
</tr>
<tr>
<td>C=16, DP=40960, F=32</td>
<td>0.124</td>
</tr>
<tr>
<td>C=4, DP=409600, F=8</td>
<td>0.258</td>
</tr>
</tbody>
</table>

SLB, ILB, and AAB registrations have been evaluated with toy examples, simulated and real datasets. Lastly, a qualitative evaluation by a pool of expert radiologists is provided, in order to assess the feedback from the final user.

Figure 12: 1D function regression results using the SLB (a, b) and ILB (c, d) techniques. Results are shown using both random samples (first column) and equally spaced ones (second column). On the bottom of each diagram fuzzy kernels shapes are plotted.
5.1. Tests on Fuzzy Kernel Regression

Before doing any test with volume data, we needed to validate the 3D Fuzzy Kernel Regression framework as a whole. In order to do this, we performed a simple mono-dimensional function regression test while evaluating the error induced by the implemented interpolation strategies that is direct interpolation from random function samples representing the landmark displacements for the SLB approach, and piece-wise linear interpolation, which has been smoothed using regression in the case of both ILB and AAB. Particularly, the AAB approach can assume the use of equally spaced landmarks that is the sub-volumes centers. Fig. 12(a) and (b) report the application of ILB using both random and equally spaced samples, while fig. 12(c) and (d) report the same task performed using ILB with its different interpolation strategy. The results reported for ILB can be assumed to hold also for AAB in the case of equally spaced samples. 16 samples were used in all the cases, and the relative fuzzy membership functions are reported on the bottom of each picture. As it can be seen, regression operated directly from the sample values leads to some oscillations, while smoothing piecewise linear interpolation leads to a much better estimation of the samples underlying function. These results were quantitatively measured computing the Root Mean Square error of the regression function w.r.t. the exact function. Results are reported in table 3 for several types of standard functions.

Table 3: Root Mean Square errors for Simple and Improved Fuzzy kernel regression, results are given both for randomly spaced ($\sigma_{dist}=1.9$) and equally spaced

<table>
<thead>
<tr>
<th>Function</th>
<th>Simple Random</th>
<th>Simple Equispaced</th>
<th>Improved Random</th>
<th>Improved Equispaced</th>
</tr>
</thead>
<tbody>
<tr>
<td>$sin(x)$</td>
<td>7.70</td>
<td>3.84</td>
<td>1.60</td>
<td>0.61</td>
</tr>
<tr>
<td>$sinc(x)$</td>
<td>5.18</td>
<td>3.50</td>
<td>1.78</td>
<td>1.15</td>
</tr>
<tr>
<td>$sigmoid(x)$</td>
<td>1.76</td>
<td>0.92</td>
<td>0.19</td>
<td>0.06</td>
</tr>
<tr>
<td>$gauss(x)$</td>
<td>4.38</td>
<td>2.83</td>
<td>1.20</td>
<td>0.56</td>
</tr>
</tbody>
</table>

As it is evident from the results, the improved method outperforms the simple one. In addition, the lower the variance of the distances between the known function samples, the lower the resulting regression error, which decreases asymptotically. Errors in simple method are mainly due to the
fluctuations occurring as getting further from the samples: the bigger the distance between two samples, the larger the fluctuations. In ILB, fluctuation is removed because the values are constrained by linear interpolation. This is a theoretical basis, which confirms that these methods can be used for recovering registration functions.

5.2. Tests on Image Registration

In order to validate the performance of the registration framework, several tests were conducted using the three proposed applications. Both synthetic and real datasets were used for the experiments. We used the Brainweb generator [35] as the source of synthetic data, while real data were obtained from the Oasis database [36] and scans provided by “Azienda Ospedaliera Ospedali Riuniti di Sciacca”. The registrations were performed using a distance-based interpolation method [37]. We used both CT and MR scans. Moreover MR volumes where PD-, T1-, and T2-weighted. The results of SLB, ILB, and AAB were compared during the experimentation.

5.2.1. Synthetic Multimodal registration

In the first experiment, we evaluated the performance of the registration schemes on synthetic mono-modality data. To generate volume pairs, we started from a single volume obtained from the Brainweb database; the volume was corrupted by 3% noise and 20% intensity non-uniformity. We then produced some artificial random deformations with a maximum amplitude of 20 voxels. This is done by means of Thin-plate Spline surfaces. Such deformations were applied to the target volume, which was registered back successively. The error was evaluated computing the average intensity differences (AID) and the root mean square (RMS) of the local registration error in each voxel. The results for this experiment are reported in fig. 13 and fig. 14.

Fig. 13 shows one slice extracted from the original volume, an example of deformation from the same slice, the average of the transformations applied to the slice, and the average of the registrations using the three proposed methods. The figure shows the same information both for the axial plane and for the coronal one. Sharper images indicate that the registrations are, in average, more accurate. In addition, in fig.14 are reported box-plot diagrams for AID and RMS indexes. The same results, along with relative computation times on a AMD Phenom Quadcore equipped with Matlab, are summarized in table 4.
Figure 13: Average of the results obtained over 200 registrations with random deformations applied to the same original volume. Results are shown both for a slice in the axial plane (a)-(f) and for another one in the coronal plane (g)-(l).
Figure 14: Box-plot of the registration results over 200 volume pairs produced by the same original one. The graphs show the distribution of the average intensity difference (AID) and the root mean square (RMS) of the registration error. Registrations were performed using the application of the proposed framework in its three versions: simple landmark based (SLB), improved landmark based (ILB) and automatic area-based (AAB).

Table 4: Results for the experiments shown in fig. 13

<table>
<thead>
<tr>
<th></th>
<th>AID</th>
<th>RMS</th>
<th>time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLB</td>
<td>9.89 ± 2.42</td>
<td>25.99 ± 4.28</td>
<td>11.70 ± 0.09</td>
</tr>
<tr>
<td>ILB</td>
<td>7.63 ± 2.34</td>
<td>21.26 ± 5.12</td>
<td>13.58 ± 0.05</td>
</tr>
<tr>
<td>AAB</td>
<td>2.18 ± 1.41</td>
<td>8.43 ± 3.35</td>
<td>165.39 ± 12.11</td>
</tr>
<tr>
<td>initial</td>
<td>11.15 ± 2.64</td>
<td>26.89 ± 5.30</td>
<td></td>
</tr>
</tbody>
</table>

5.2.2. Sequential registration

This experiment regards scans from the same patient that have been acquired at different times, and present anatomical differences (for example due to resections). Tests were operated on 30 real CT case studies. Fig.15 shows an example for both the axial and the coronal plane, using AAB registration. As it can be seen in this case study, the pose misalignment are diminished providing a complete overlap of the target volume and the registered one (fig.15e and 15g). A checkerboard visualization has been used to show the effectiveness of registration (fig.15f and 15h) both in the ventricles and other anatomical structures. The results have been compared to MIRT (fig.15i and 15k). As it can be seen the result of the alignment is comparable, little misalignments occur in both images, but in different places.
Figure 15: Registration results for different time inter-patient scans. Floating slice extracted from the volume in the axial and coronal plane (a,c) and reference slice (b,d) are shown in first row. Registration result for AAB (e,g) and MIRT (i,k) are shown in second row along their respective checkerboard overlap (f,h) and (j,l).
5.2.3. Real multimodal registration

The last registration experiment is related to real multi-modal images. Volumes acquired with different technology equipments were involved in the registration process. Inter-patient images with extremely different anatomies were used to stress the robustness of the system. The tested cases can present also pathologies or diseases, which vary drastically the intensity levels distribution across the volume. An example of such registrations is shown in Fig.16 for AAB. The reader should note the different head shape and the stain (fig.16a-b). AAB (fig.16c) allows free-form like transformations and deforms the structure to achieve successful alignment of the whole anatomy. Result are compared to MIRT registration strategy (fig.16d). The checkerboard visualization is used to show the effectiveness of the registration both for AAB and MIRT (fig.16e-g).

Figure 16: CT-T2 registration example test. In the first row, a slice taken from one of the tested volumes (a) is registered onto a target slice (b) taken from a volume of a different patient, which has been acquired with a different modality. The two slices are on the axial plane. Registration is performed with both AAB (c) and MIRT (d). From (e) to (g): checkerboard visualization of the initial overlap, the AAB overlap, and the MIRT result.
5.3. Expert validation

In order to add value to the proposed framework and its applications, we submitted our results to a team of expert radiologists for evaluation. We believe that such activity is very important for establishing whether results are satisfactory not only from a numerical or visual perspective, but also for using the system on real diagnosis tasks. The evaluation was taken by 4 radiologists and it is based on the following levels of assessment:

- **Global alignment rating**: overall evaluation of the registration procedure, in terms of shapes and contours matching.
- **Availability of the points of interest**: this defines how many points of interest are available in the very same positions of the images. For each test case a list of features is given and their alignment after registration is rated.
- **Morphological structures coherency**: the coherence of the morphology after the volumes registration is rated in order to report any structure deformation or anomaly.

For the evaluation two kinds of assessments were taken: first, the whole anatomical volume registration performance was ranked. 12 test cases were considered, each one consisting in a pair of MR datasets (T1-, T2- or PD-weighted). Results for these tests are reported in Table 5, and some examples are depicted in Fig. 17. The score represents how much the registered volumes and/or slices are useful to the expert for her/his analysis. It is an absolute value that can be interpreted on the following scale:

- 1: No use at all.
- 2: Poor informative content making very hard to make any assessment.
- 3: Average informative content, giving some cues for the assessment.
- 4: Good registration, a fair analysis can be taken.
- 5: Very good registration, analysis is radically improved.

A second evaluation was taken considering slice-wise comparisons to assess just some anatomical points of interests. 30 test cases were used for the evaluation. Each test case was a pair of T1-, T2- or PD-weighted MRI slices taken from different planes. Results are reported in table 6. In both cases, in addition to the existing misalignment, further random rotation and translation has been injected to stress the method. In the two tables, each row reports the test number, the compared slices type if applicable, the precise points of interest to be investigated if applicable, and the rating for the three levels of assessment ranging from 1 to 5. Some of the low-rated test cases came out to be outliers, since after the artificial rotation and translation some
Table 5: Volume expert evaluation for the registration procedure. Vote ranges from 1 (minimum) to 5 (maximum).

<table>
<thead>
<tr>
<th>Test #</th>
<th>Global align.</th>
<th>Pts.of interest</th>
<th>Morph.coherency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Mean</td>
<td>3.67</td>
<td>3.50</td>
<td>3.50</td>
</tr>
</tbody>
</table>

Figure 17: Two examples of registered time-sequential volume scan submitted to the radiologists for evaluation. Note that some elements in the floating volume are missing in the reference one and vice versa.

Peripheral information was out of the cropping bounds. Consequently, some of the structures got lost and were not realigned. Even if just some local regions were aligned correctly, the whole registration task was compromised, thus diminishing the score.
Table 6: Expert evaluation for the registration procedures

<table>
<thead>
<tr>
<th>Test #</th>
<th>Slice type</th>
<th>Points of interest under investigation</th>
<th>Global align.</th>
<th>Pts.interest</th>
<th>Morph.coherency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Axial T1 vs T1</td>
<td>Lateral v., corpus c., frontal g.</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Axial T1 vs T1</td>
<td>Lateral v., 3rd v., basal ganglia, thalamus</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>Axial T1 vs T1</td>
<td>4th v., bulb, cerebellar hemisp., maxillary sinus</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Coronal T1 vs T1</td>
<td>Lateral v., basal ganglia, Sylvian fissure, opt. chiasm and tract</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>Sagittal T1 vs T1</td>
<td>Temporal g., cerebellum</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>Axial T1 vs T1</td>
<td>4th v., bulb, cerebellar hemisp.</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>Axial T2 vs T1</td>
<td>Lateral v., corpus callosum, frontal g.</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>Axial T2 vs PD</td>
<td>Acqueduct, midbrain, basal cistern, cerebellar hemisp.</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>Axial T2 vs T1</td>
<td>Lateral v., basal ganglia, thalami</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>Axial T2 vs PD</td>
<td>4th v., bulb, cerebellar hemisp., maxillary sinus</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>Axial T1 vs T1</td>
<td>Fronto-parietal g.</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>12</td>
<td>Axial T2 vs T1</td>
<td>Basis pontis, 4th v., acoustic nerve, cerebellum, ocular bulbs</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>13</td>
<td>Axial T2 vs PD</td>
<td>Basis pontis, 4th v., acoustic nerve, cerebellum, ocular bulbs</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>14</td>
<td>Coronal T1 vs T1</td>
<td>Frontal g., orbital fat</td>
<td>4</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>Coronal T1 vs T2</td>
<td>Frontal g., orbital fat</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>16</td>
<td>Coronal PD vs T2</td>
<td>Parieto-occipital g., cerebellum, sup. sagittal sinus</td>
<td>4</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>17</td>
<td>Sagittal T1 vs T2</td>
<td>Temporal lobes, cerebellum</td>
<td>5</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>18</td>
<td>Sagittal T1 vs T1</td>
<td>Temporal lobes, cerebellum</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>19</td>
<td>Sagittal T2 vs T2</td>
<td>Occipital horn, cerebral g., cerebellum</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>20</td>
<td>Sagittal PD vs T1</td>
<td>Occipital horn, cerebral g., cerebellum</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>21</td>
<td>Axial T1 vs T1</td>
<td>Midbrain, Cerebellar vermis, temporal g., ocular bulbs, opt.nerves</td>
<td>3</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>22</td>
<td>Sagittal T1 vs T1</td>
<td>Fronto-temporal g., cerebellum, maxillary sinus</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>23</td>
<td>Sagittal T1 vs T1</td>
<td>Corpus callosum, fornix, brainstem, cerebellum, pituitary</td>
<td>5</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>24</td>
<td>Sagittal T1 vs T1</td>
<td>Corpus callosum, fornix, brainstem, cerebellum</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>25</td>
<td>Axial T1 vs T1</td>
<td>4th v., temporal lobes, cerebellum</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>26</td>
<td>Axial T1 vs T1</td>
<td>Brainstem, temporal lobes, cerebellum</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>27</td>
<td>Coronal T1 vs T1</td>
<td>cerebellum, occipital g.</td>
<td>3</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>28</td>
<td>Coronal T1 vs T1</td>
<td>cerebellum, occipital g.</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>29</td>
<td>Sagittal T1 vs T1</td>
<td>Corpus callosum, brainstem, cerebellum</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>30</td>
<td>Sagittal T1 vs T1</td>
<td>Fronto-temporal g., cerebellum, maxillary sinus</td>
<td>5</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Mean</td>
<td>-</td>
<td>-</td>
<td>4.57</td>
<td>4.97</td>
<td>4.40</td>
</tr>
</tbody>
</table>
6. Conclusion and future works

In this paper the 3D Fuzzy Kernel Regression framework for non-rigid image/volume registration was presented. After discussing the theoretical foundations of the presented approach, three different applications were presented and compared with state-of-the-art registration techniques. The framework relies on kernel regression and fuzzy c-means. In particular, fuzzy membership maps obtained as the result of fuzzy clustering are used as equivalent kernels for regression. In turn they allow to recover all the deformation parameters needed to align the input volume to the target one, on the basis of some prior information on the volumes pair. Three applications of the framework have been introduced and discussed. The first two registrations (SLB and ILB) are landmark-based approaches, while the third one (AAB) is area-based. SLB and ILB rely on either manual selection or automatic extraction of some landmark points to recover the transformation in a one-shot fashion, while AAB needs only the two volumes as its input, and recovers the needed correspondences (and the consequent local deformations) by maximizing the normalized mutual information of the volumes’ sub-regions.

The parallel implementation of the proposed applications by means of CUDA-enabled GPU devices has been reported. Parallelization is easily achievable due to the intrinsic local data independence of the framework where some computations can be carried on separately for each region. Joint gray level histogram computation and the FCM algorithm have been implemented on GPU. The former is used to compute NMI, while the latter is used to compute the fuzzy membership maps used in the regression process.

The framework and its application were tested and evaluated with several experiments. At first, the value of fuzzy kernel regression was assessed from a purely analytical perspective by measuring the error introduced when reconstructing functions from a limited subset of samples. Both equally- and unequally-spaced samples are considered and the differences in the results are reported.

The second experiment was conducted onto synthetic brain volumes for which the actual registration ground truth is known. Volumes are artificially deformed and then registered back. Averaging the results for 200 datasets, and extracting single slices in whatever plane of a volume, gives images whose amount of blurring depends on the goodness of the achieved registration. Qualitative and quantitative results are also reported.

Additional evaluation tests were operated on both mono- and multi-modal
real datasets. Mono-modal tests were conducted for intra-patient evaluation, with scans taken at different time and presenting anatomical differences. For multi-modal volumes different patient scans were used to align completely different anatomical structures. Comparisons with the MIRT state-of-the-art method is provided.

Finally, real scans have been evaluated by a pool of 4 expert radiologists to assess the performance of the proposed technique as regards its effectiveness as a support of the diagnostic process.

Although, the framework has been presented for addressing a hot topic in medical imaging, it can be regarded as a general tool for pattern recognition in image and video processing. Future work will be devoted to investigate the applicability of the framework in video segmentation and tracking. The main idea is to “register” some consecutive frames in a video stream to lock an object of interest, and to perform subsequent registrations with incoming frames to track the object in the scene. The object of interest could be also registered at first with a template of a “known” object, which has to be (re-)identified in the video.

References


