2.3 Correlation as a measure of matching

The former operator is highly suitable for CT–MR landmark extraction because it gives a high response in the voxels depicting the skull, which yields to a quick and accurate alignment of the images. The first step is to reformat the source images into cubic voxels. After the creaseness extraction, the following step is to iteratively transform one of the images until it becomes properly aligned with the other.

From here on, we will normally refer the two images as static S and dynamic D, because in the optimisation step the first is taken as model, and the second is transformed iteratively until it is aligned. Signal processing literature also employs the nomenclature f and g.

A suitable function to measure the quality of the alignment is the correlation function

$$C_T = f(\vec{x}) \circ g(T(\vec{x})) = \sum_{\vec{x} \in f} f(\vec{x}) \cdot g(T(\vec{x})) \tag{2.21}$$

where f and g are the creaseness images and T represents a transformation whose parameters we want to assess. A key step is not to transform all the pixels in the image, but only those with values higher than a small fixed threshold. This saves up to 95% of the total computations, due to the sparcity of creases in an image.

Since we are employing the correlation as a measure of alignment, it is interesting to check the properties described on page 7. For the sake of simplicity, we consider only 1 dimensional images, with positive values: $S = \{s_1, \cdots, s_n\}$, $D = \{d_1, \cdots, d_n\}$:

- **continuity**
  For the nearest neighbour scheme, this measure would not be continuous:

  $$\lim_{h \to 0^-} (\mathbf{f} \star \mathbf{g})(\alpha + h) \ = (\mathbf{f} \star \mathbf{g})(\alpha - 1) \ \neq \lim_{h \to 0^+} (\mathbf{f} \star \mathbf{g})(\alpha + h) \ = (\mathbf{f} \star \mathbf{g})(\alpha)$$

  The correlation measure would be continuous for the linear interpolation scheme, since transformations computed with this interpolation observe:

  $$\lim_{h \to 0^-} \mathbf{f}(\alpha + h) \ = \lim_{h \to 0^+} \mathbf{f}(\alpha + h) \ = \mathbf{f}(\alpha)$$

- **nil element**
  We demonstrate that the maximum of the correlation of an image with itself occurs at a null translation; other maximum with same values for other translation may exist, i.e., the inequality is not strict.

  We compare the value of correlation with a null translation with the value of correlation obtained with a translation of 1 pixel, this is,

  $$\sum s_i^2 \geq \sum s_i s_{i+1} \tag{2.22}$$

  We will proceed by induction. First we prove 2.22 for $n = 2$:

  $$\begin{aligned} s_1^2 + s_2^2 \geq s_1 s_2 + s_2 s_1 = 2 s_1 s_2 \qquad & \Leftrightarrow \\ (s_1 - s_2)^2 \geq 0 & \end{aligned}$$

which is true for any values of s_1 and s_2. For convenience in annotation, we define $s_{n+1} \equiv s_1$

.

Next step is to suppose equation 2.22 true for $n-1$ values:

$$s_1^2 + s_2^2 + \cdots + s_{n-1}^2 \geq s_1 s_2 + s_2 s_3 + \cdots + s_{n-2} s_{n-1} + s_{n-1} s_1$$

We have to prove the general case for n values:

$$s_1^2 + s_2^2 + \cdots + s_n^2 \geq s_1 s_2 + s_2 s_3 + \cdots + s_{n-1} s_n + s_n s_1$$

We will proceed by variable substitution. Let $s_m = \min_i(s_i)$. Then, we write all s_i in terms of s_m:

$$s_i = s_m + \tilde{s}_i, \qquad i = 1 \cdots n$$

We develop the left term of the equation (2.22):

$$\sum_{i=1}^{n} s_i^2 = \sum_{i=1}^{n} (s_m + \tilde{s}_i)^2 = n s_m^2 + \sum_{i=1}^{n} 2 s_m \tilde{s}_i + \sum_{i=1}^{n} \tilde{s}_i^2$$

And then the right term:

$$\sum_{i=1}^{n} s_i s_{i+1} = \sum_{i=1}^{n} (s_m + \tilde{s}_i)(s_m + \tilde{s}_{i+1}) = n s_m^2 + \sum_{i=1}^{n} 2 s_m \tilde{s}_i + \sum_{i=1}^{n} \tilde{s}_i \tilde{s}_{i+1}$$

Replacing and simplifying:

$$\sum_{i=1}^{n} \tilde{s}_i^2 \geq \sum_{i=1}^{n} \tilde{s}_i \tilde{s}_{i+1}$$

$\tilde{s}_m = 0$ from the definition of s_m. We rewrite last equation taking that into account, and assuming $m = n$, i.e. the last s_i is the minimum, in order to simplify notation:

$$\tilde{s}_1^2 + \cdots + \tilde{s}_{n-1}^2 + \overbrace{\tilde{s}_n^2}^{0} \geq \tilde{s}_1 \tilde{s}_2 + \cdots + \tilde{s}_{n-2} \tilde{s}_{n-1} + \overbrace{\tilde{s}_{n-1} \tilde{s}_n + \tilde{s}_n \tilde{s}_1}^{0}$$

However, for $n-1$ we stated

$$s_1^2 + \cdots + s_{n-1}^2 \geq s_1 s_2 + \cdots + s_{n-2} s_{n-1} + s_{n-1} s_1$$

Taking $s_i = \tilde{s}_i$ and since $\tilde{s}_i \geq 0$ by construction:

$$\tilde{s}_1^2 + \cdots + \tilde{s}_{n-1}^2 \geq \tilde{s}_1 \tilde{s}_2 + \cdots + \tilde{s}_{n-2} \tilde{s}_{n-1}$$

Which makes the statement true the general case, and proves the inequality.

For general transformations, we just need to consider that linear interpolation does not introduce values higher than the original.

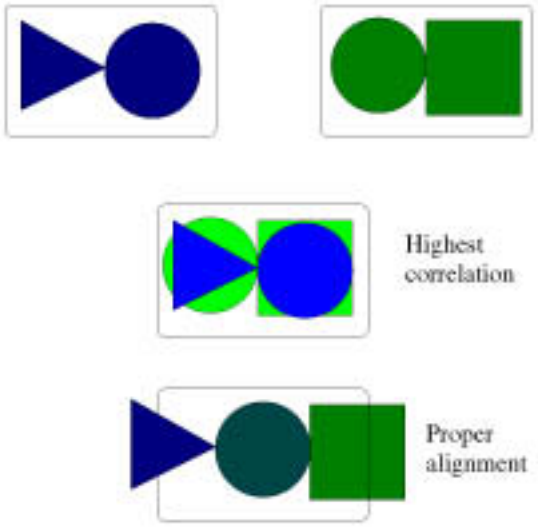

Figure 2.10: The highest correlation value is not at alignment

- **monotonicity**

  For general images, the correlation does not grow monotonically until images are aligned. This undesirable property will motivate further discussion in following sections.

The correlation function is a common topic in image processing books. Refer, for instance, to [90] or [23, pg 67] for more information.

In short, these properties warranty that the alignment function is suitable for registration:

- The maximum is achieved when the two images are aligned.
- It provides sub-pixel accuracy.
- The profile is smooth.
- Compared to normalised correlation, the linear correlation needs only one pass per pixel.

The correlation value for a pair of images, one transformed iteratively, can be computed in two fashions. The straightest is to transform the dynamic image and then compute the correlation, but, as stated previously, the computational time can reduced up to 95% by joining the two operations. Also, the correlation computed with this scheme has the additional advantage that the intermediate transformed image does not need to be stored. Figure 2.11 describes its pseudo-code.

The only advantage of the normalised correlation is that the measure is independent of the grey value of the images; linear correlation is higher for images with higher

Given two images S and D, a transformation T to apply to D and a threshold value c, the scheme to compute the correlation is

$cor = 0$
For each pixel Si, j in S,
  if $S(i, j) > c$
    Find the corresponding coordinates in D: $(k, l) = (i, j) \star T$
    Interpolate the value $D(k, l)$ using linear interpolation with neighbouring values.
    $cor+ = S(i, j)D(k, l)$

Figure 2.11: Algorithm to reduce the computational cost of correlation

values. The linear correlation value depends not only on the goodness of the alignment, but also on the number and value of the pixels. This is relevant only when comparing the final alignment of registration of different pairs of images.

The cross-correlation does not give for all cases the same response as a human operator would. Examples can be build were the maximum given the function differs from the human alignment concept, because the later considers more concepts than strictly the number of pixels aligned. See, for instance figure 2.10.

2.4 The hierarchical approach to optimisation

In previous sections we have showed that the creaseness operator is a suitable choice to segment the skull, and that the correlation has good properties as a measure of alignment. In this section we give full report of the technical details of our algorithm.

2.4.1 Building the pyramid

Initially we have considered only the rigid type of transformation, which includes 3 translation parameters (one for each axis), plus 3 rotation parameters (one for each axis). The rotations are taken at the centre of the image, because the angles obtained in this fashion are smaller than the using the top left corner as centre of rotation. Smaller angles are more likely to be found because the initial search samples only a short range of angles. The full specification of the transformation matrix is included in appendix A

The function C_T together with the 6 parameters of the transformation defines a search space which is difficult to optimise because:

1. the function is non–monotonic, i.e. has many local maxima.

2. the similarity measure is expensive to compute since it involves the transformation of a large 3–D image plus a product of two images.

3. translation and rotation parameters can not be decoupled in order to reduce the dimension of the search space, as it can be done in 2–D [91, 1, 41, 4].

An approach to overcome the two first problems is to search within the parameter space at multiple resolutions. As in [102] we handle multiple resolution by building two pyramids where the CT ridgeness and MR valleyness images are at the bottom and each level is a sampled version of the previous at half resolution, until images have a final size of about 16 pixels in each dimension.

Such a scheme has been largely employed. Usually, the image is sampled after blurring with a Gaussian, with the aim of avoiding that the sampling would enlarge the noise of the image. But, in our case, this sampling would not be interesting since it would spread our precise segmentation, thus becoming useless. Therefore, we simply take for each pixel at one level the maximum of a local neighbourhood from the previous level. This approach preserves information through the hierarchy.

We have designed an experiment to check whether the shape of correlation function is preserved through the levels of the pyramid. We have registered two CT and MR volumes, and then sampled the correlation function at intervals at the function maximum. We have seen, as figure 2.12 shows, that the profile is approximately the same, and therefore our approach is valid. In addition, since the shape is actually smoother, the optimisation will less likely miss the maximum, thus being more robust.

2.4.2 Exhaustive search

For the finest resolution, an exhaustive search would not be possible because the number of operation would have been too large. The search starts at the top of

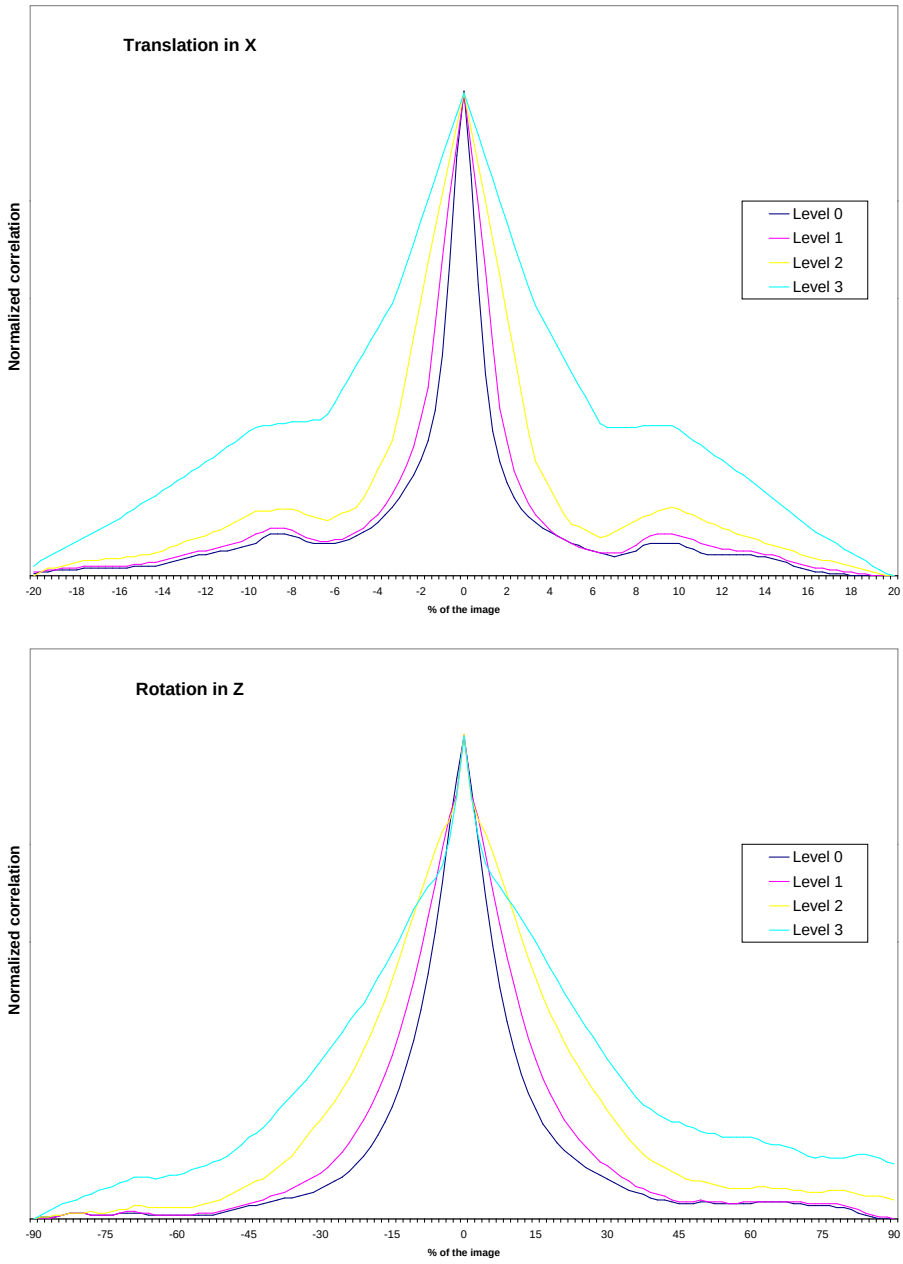

Figure 2.12: Correlation values (y axis) sampled at registration (centre in the x axis), and different intervals of x translation (Top) and z-rotation (Bottom). Throughout the pyramid hierarchy, the function profiles is smoothed but its shape is preserved.

pyramid, where the small size of the images permits an exhaustive search. At this level we are not interested in accurate results, but in obtaining a broad set of promising seeds.

The first approach was to iterate each transformation parameter at a number of steps. But that rapidly leads also to a high number of trial values. It is much more efficient to operate in the Fourier domain. The shift theorem permits to compute n^3 trial translation at the computational cost of a single multiplication. Using the approach, only the rotation parameters need to be iterated.

$$f(\mathbf{x}) \circ g(\mathbf{x}) \quad \Leftrightarrow \quad F(\mathbf{u})G^*(\mathbf{u}) \tag{2.23}$$

$$f(\mathbf{x})g^*(\mathbf{x}) \quad \Leftrightarrow \quad F(\mathbf{u}) \circ G(\mathbf{u}) \tag{2.24}$$

We had two problems when implementing the Fourier computations. The first was that images needed to have the size power of two, to make use of the Fast Fourier implementation. Also, if images were not cubic the transform to the Fourier domain produced phantom values which misleaded the search. The same problem appeared when tried to rotate the images at the Fourier domain. Taking into account these limitations, we finally employed the algorithm set in 2.13.

Given two images S and D and a set of rotations to be sampled,
Build two cubic images S_C and D_C with size power of two able to include them.
Insert the images S and D into S_C and D_C, located at the centre.
Compute the Fourier transform of S_C, $S_F = FFT(S_C)$.
For each rotation r in the set,
  Rotate the dynamic image, $D^r = r(D_C)$
  Compute the Fourier transform, $D_F^r = F(D^r)$
  Compute the product $SD = S_F \star D_F^r$
  Anti-transform the product $SD^{-1} = FFT^{-1}(SD)$
  The values of SD^{-1} are the correlation values needed.

Figure 2.13: Algorithm to compute the correlation at the Fourier domain

Table 2.1 gives the statistics of computation times for several operations at each level of the pyramid. These times, set for an image with size (334×334136), are similar for the other images. The time corresponding to level 0 was skipped because the cubic image ($512 \times 512 \times 512$) was too large to fit in memory.

Often the best transformation at any level is not the one which later will lead to the final solution, because the hierarchical approach introduces false maxima. Therefore, for the sake of robustness we have chosen to keep several values from each level as seeds for the next. For each level we reduce the number of seeds to the half, to end optimising a single value at the highest resolution level. The third column in 2.1 is the time for each rotation to be sampled, computed from the sum of the times in figure 2.13: a rotation, an FFT, and an inverse FFT. Since there are three rotation parameters, the total time for a through sampling can be very large.

Level	Size (pixels)	FFT	Transform. Time (sec)	Full Iteration
1	256	†	41.84	
2	128	17.2	2.03	36
3	64	1.7	0.26	3.6
4	32	0.14	0.03	0.31

Table 2.1: Computation times taken by the Fourier transformations.
† Not enough memory to compute it.

We have chosen the following values for the exhaustive search: we sample the pyramid to level 4, we take 10 rotation values for each axis, and the step value is 0.1 rad. The total computation time for this search is about 5 minutes. Probably the time could be reduced by taking less steps for the exhaustive search, but the price again would be a decrease in the final robustness.

2.4.3 Iterative optimisation

The initial exhaustive search has provided us with a set of transformations, which are maxima in the correlation for the images in the sampled pyramid. The next step is to refine these values through the hierarchy, until, at the finest resolution, the convergence leads to the highest value.

We have used the downhill Simplex algorithm as implemented in [80] to maximise the correlation function at all levels except the initial. Each search, started from the seeds from the previous level, finishes when the algorithm achieves a zone where the difference between the maximum and the minimum values found in the neighbourhood is lower than a threshold. The tolerance value and number of seeds in the highest resolution level determine to a high degree its robustness and final time.

We have implemented the optimisation search also with a Simulated Annealing and Powell's algorithms, but they did not improve the results as a rule. This is easy to explain because we did not consider the order of convergence, but rather whether it converged or not, which is already taken into account by the hierarchical structure. For medical volume image registration, the total time is not very relevant as far as it keep reasonable, this is, below 15 minutes. It is far more important to ensure robustness.

Also, we have extended the matching scheme with chamfer distances as described in [27]. This algorithm assigns to each voxel the approximate distance to a surface, i.e. the segmented crease, by means of a fast correlation of the image with two masks. We apply this process only to the first and second levels (highest resolution) of the CT image, and at the following levels the optimisation steps run without modification.

As a result, the correlation function does not seek the exact matching of the two surfaces, but also their proximity, and the peak of the function corresponding to the maximum becomes wider and easier to locate. In the following section we present the experiments to validate its use.

Level	Dim (x,y)	Slices	Cor	Cor (thr)
0	334	136	14.5	0.52
1	167	68	1.8	76E-3
2	83	34	0.24	10.5E-3
3	41	17	0.026	165E-6
4	20	17	0.0062	61.6E-6

Table 2.2: Computation times in seconds taken by the correlation process

2.4.4 Initial alignment

Another important issue is the initial alignment of the two volumes, specially when their field of view differs. Although both show the full boundary of the skull in the axial axis, the number of slices provided for one modality sometimes is the double of the other one. See, for instance, figure 2.14 showing an example taken from the Vanderbilt's database.

Despite this fact, at the beginning a very simple procedure consisting in aligning their respective centre of mass worked well. Once we had to apply the registration to the large database from Vanderbilt's, we realised some failures were due to this cause. For a very bad initial choice, the short image stack would be placed too close to a non-corresponding surface, and it would fail to converge. Whereas, for the image being placed at a reasonable choice, the algorithm worked properly.

Although this problem happened only to a single pair of images, it hindered us to consider our method as fully automatic. We needed a procedure to do an automatic coarse alignment of the images, and we decided to make use of the geometric features of the head. In an axial view, slices depict an elipsoidal form above the eyes, and a square-like form below it. Thus, a compact representation is the list of the principal axes of each slice.

We work with the original modality images because in this manner the procedure is independent of missing segments in the segmentation step. For each slice, we separate the head from the background with a simple threshold, then we compute the length of the longest segment for each axis. After this procedure, very fast, we obtain for each volume a set of pairs $a_i = \{a_x, a_y\}$ and $b_i = \{b_x, b_y\}$ of length m and n, m and n not necessarily the same.

The next step is to estimate the translation in the z coordinate which best aligns the images. We measure the alignment between the two sets with:

$$\text{Align}(d) = \sum_i \parallel a_i - b_{i-d} \parallel \tag{2.25}$$

and simply take the translation d with lowest value. This measure is robust against outliers caused by wrong extractions of slices, which may happen when artifacts appear in the image.

$$T_z = \text{Arg}\min_d \ \text{Align}(d) \tag{2.26}$$

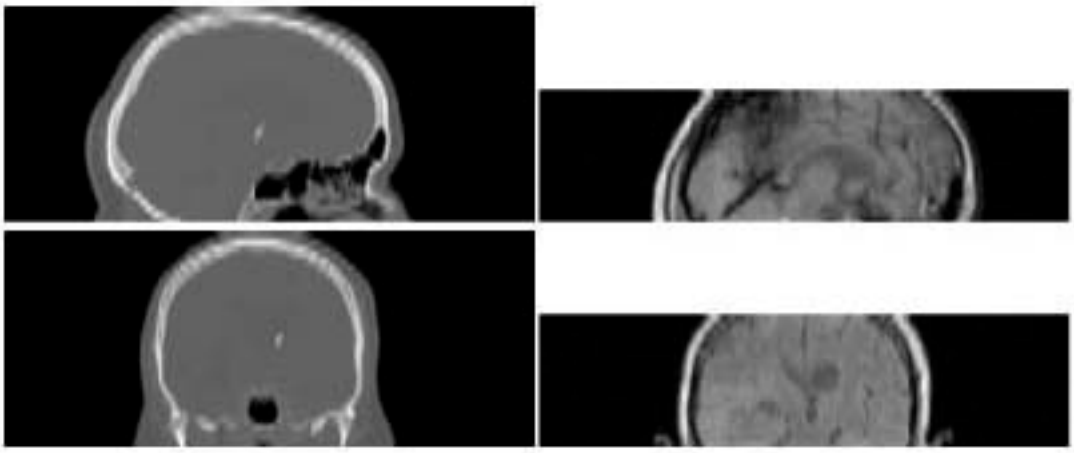

Figure 2.14: Sometimes images do not overlap because one image is a partial volume of the other. This example, taken from the Vanderbilt database, shows sagittal (top) and coronal (bottom) view of CT image (left) with the MR image (right) scanning only central slices.

See figure 2.17 for a sample of the images at each step before to the actual optimisation starts. This preprocessing is necessary to ensure a proper convergence when the field of view of each volume is very different.

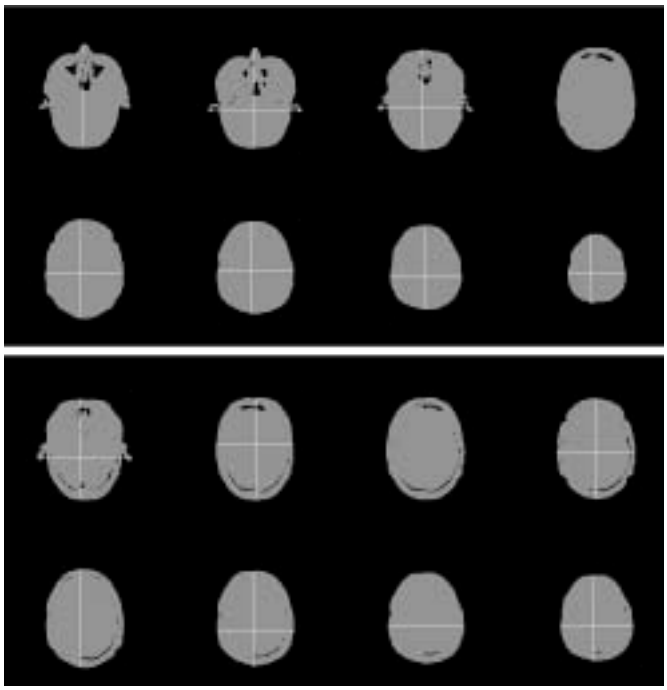

Figure 2.15: Detecting the main axis of the CT image (top) and MR image (bottom) to compute the initial alignment.

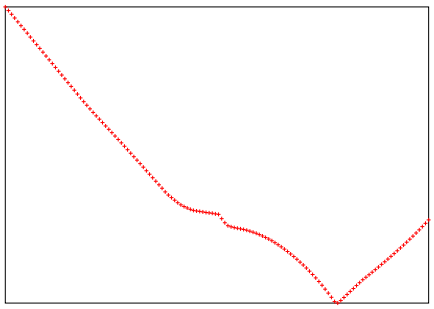

Figure 2.16: The initial translation in the z axis is estimated robustly by comparing the main axis of the head at each slice. The comparing function, as defined in equation 2.25, has a single minimum.

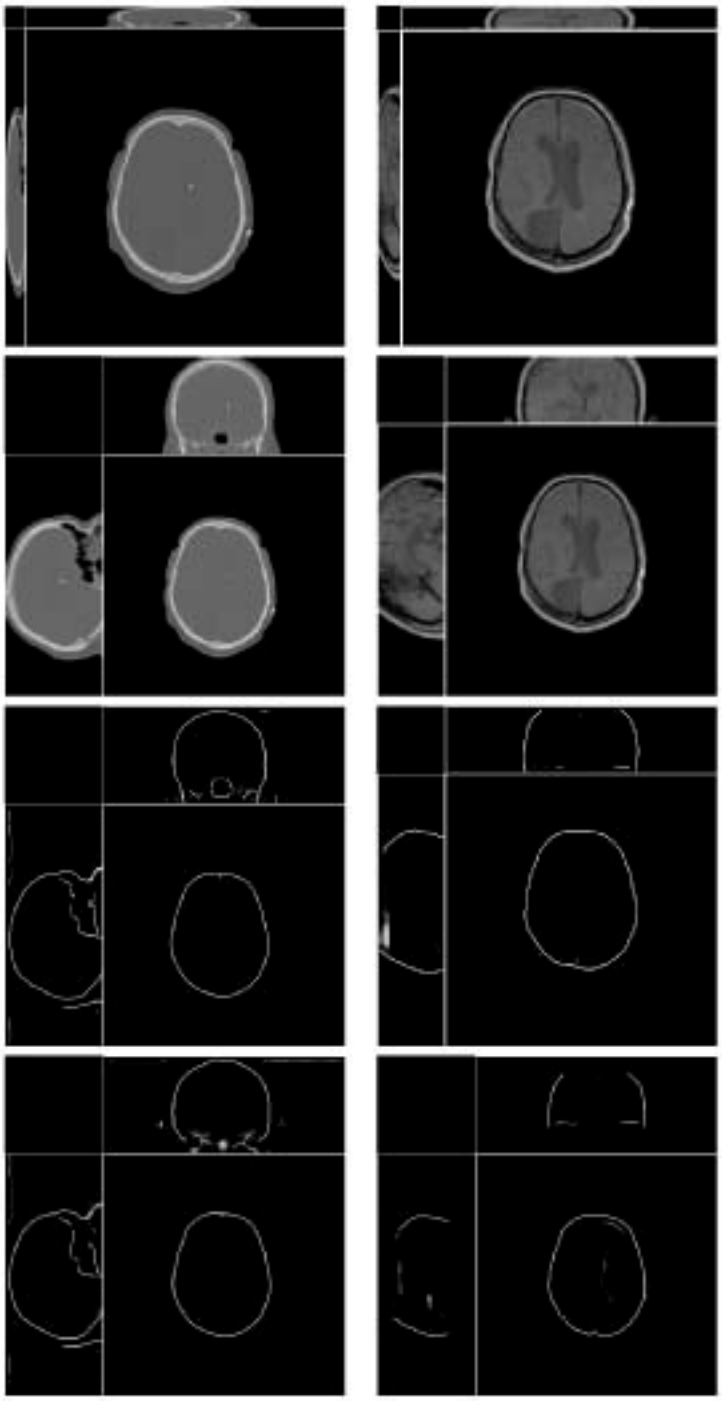

Figure 2.17: Steps before images are set to register, from top to bottom: original, formatted at 1.0 pixel size, creaseness images and crests after estimating the centres.

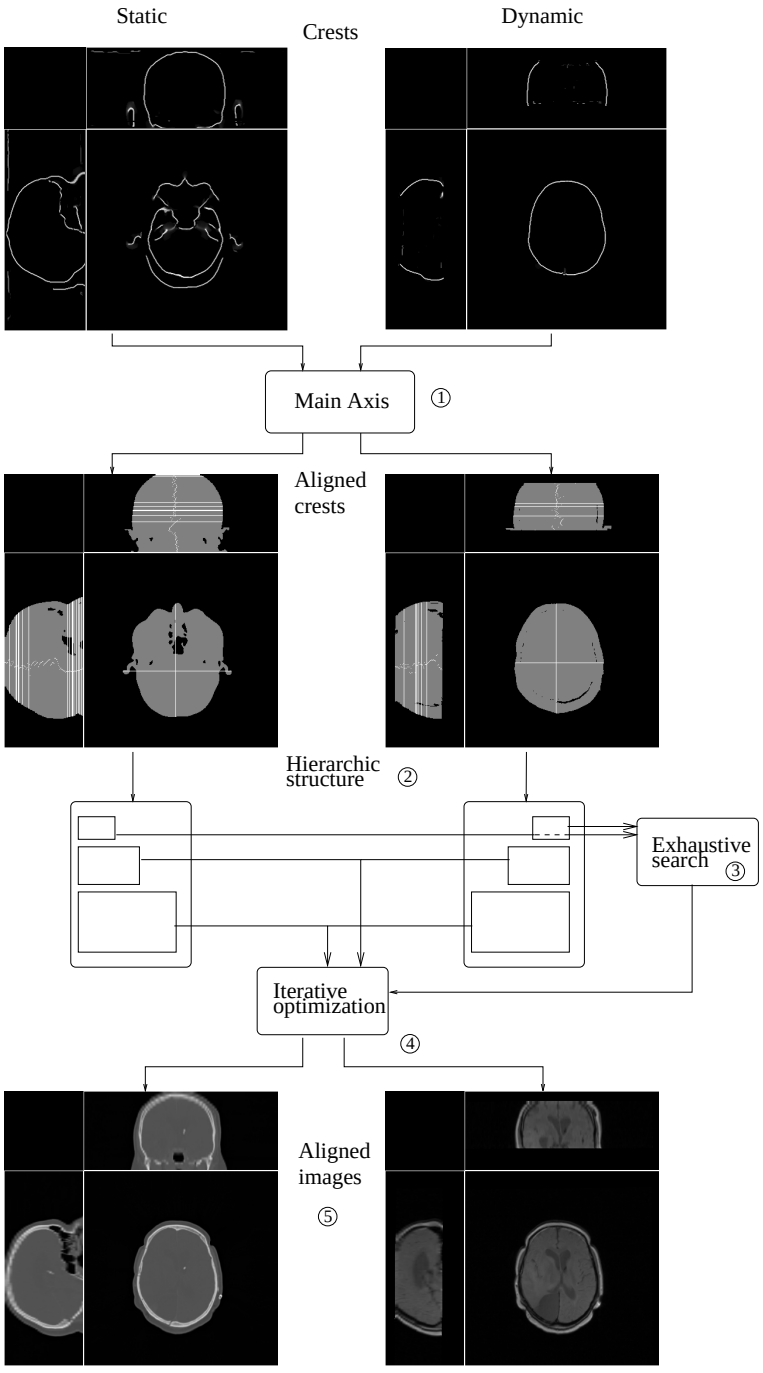

Figure 2.18: General scheme of our registration algorithm