Medical Image Processing

7 Registration

Prof. Dr. Marcin Grzegorzek[1]

Research Group for Pattern Recognition
Institute for Vision and Graphics
University of Siegen, Germany

../PR-Logo.jpeg

---

[1] These materials have been created with a great help of M.Sc. Cong Yang.

Overview

7.1 Fusing information

The main problem lies in the fact that the reference coordinate system of a volume dataset does not reference the patient-it references to the scanner. Changing patient position between two imaging sessions is, however, necessary in many cases. Figure **??** gives a sample of the capabilities of a modern PET/CT system, and PET/MR systems are subject to clinical investigation.

Fusion of images - that is, finding a common frame of reference for both images - is therefore a vital task for all kinds of diagnostic imaging.

7.1 Fusing information

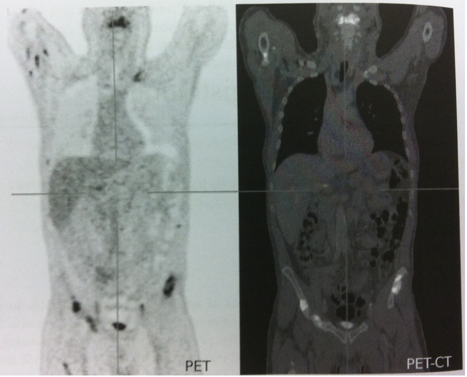


Figure 7.1: An image sample from a combined PET/CT system. Registration of the PET and the CT volume is automatically achieved by taking both volume datasets in a scanner that is capable of recording both tracer concentration and x-ray attenuation.

Overview

7.2 Registration paradigms

In general, registration algorithms determine a volume transformation from identifying common features in two coordinate systems. These features can be intrinsic(that is features that are inherent to the image data) and extrinsic.

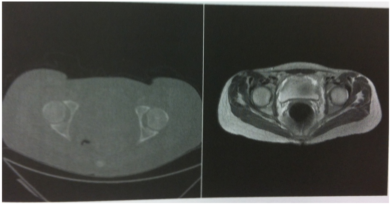


Figure 7.2: Two slices of the same patient at the same location, the hip joint. The left image shows the CT image, the right image is taken from the MR scan. Differences in the field-of-view and patient orientation are evident. Registration algorithms compensate for these discrepancies by finding a common frame of reference for both volume datasets.

7.2 Registration paradigms

7.2.1 Intra- and intermodal registration

In intramodal registration, we are registering images that stem from the same modality; an example is CT-to-CT registration of volumes acquired at different times. Such a procedure is extremely helpful when doing time series evaluation, for instance when tracking the effect of chemo- or radiotherapy on tumor growth.

When fusing image data from different modalities, we are dealing with intermodal registration. A classical example is MR/CT fusion; an example of MR/CT fusion using an similarity measure named normalized mutual information(NMI) in AnalyzeAVW can be found in Figure **??**.

7.2 Registration paradigms

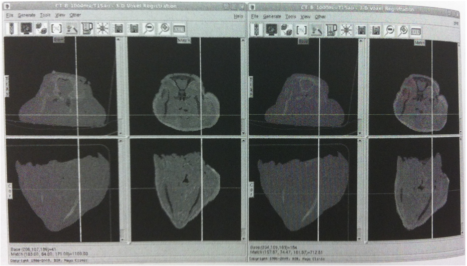
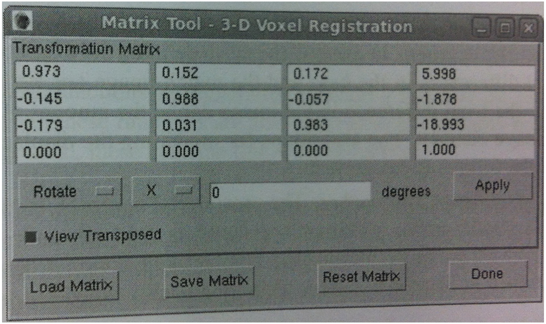

File Generate Tools View Other
Help
Base
Match
Copyright 1986-2009, BIR, Mayo Clinic

Figure 7.3: Two screenshots from the registration tool of AnalyzeAVW. On the left hand, we see two slices of the MR and the CT scan, unregistered. Gross orientation is similar, but discrepancies are clearly visible. Besides soft tissue deformation, it is also evident that rigid structes such as the calvaria of the well-known pig scan do not coincide. After initiating the registration algorithm, the two images match. The resulting volume transformation matrix V can be seen in Figure **??**.

7.2 Registration paradigms



Transformation Matrix

0.973	0.152	0.172	5.998
-0.145	0.988	-0.057	-1.878
-0.179	0.031	0.983	-18.993
0.000	0.000	0.000	1.000

Rotate X 0 degrees

■ View Transposed

Figure 7.4: The volume transformation as produced by the AnalyzeAVW.

7.2 Registration paradigms

7.2.2 Rigid and non-rigid registration

Rigid registration refers to algorithms that are confined to finding an affine transformation in three or six dof; In non-rigid registration, the affine transformation is only a first step in alignment of volumes. The algorithms for modelling the displacement field can me put into the following categories:

- Featurelet-based approaches.
- B-Spline-based approaches.
- Linear elastic models.
- Viscous fluid models.
- Finite-element models.

Overview

7.3 Merit functions

In general, merit functions in registration provide a measure on the similarity of images; therefore, these functions are also referred to as similarity measures or cost functions. In general, these have the following properties:

- They yield an optimum value if two images are aligned in an optimal manner.
- The capture(or convergence) range of the merit function should be as wide as possible.
- Merit functions for image fusion can be defined as intensity-based, or as gradient-based.

7.3 Merit functions

A number of intensity-based merit functions were defined, and most of them are basically statistical measures for giving a measure of mutual dependence between random variables. In these cases, the intensity values ρ are considered the random variables, which are inspected at the location of image elements which are assumed to be the same. The *sumofsquareddifferences* :

$$M_{SSD} = \frac{1}{N} * \sum^{N} (\rho Base(x, y, z) - \rho Match(x, y, z))^2 \qquad (1)$$

Where N is the total number of pixels. The inner mechanics of this measure are evident.

7.3 Merit functions

Another possibility to compare intramodal images is to take a look at the difference image I_{diff} of two images, and to assess the disorder in the result. Such a measure is pattern intensity, given here only for the 2D case:

$$\begin{aligned} M_{PI} = \sum_{x,y}^{N} \sum_{d^2 \leq r^2} \frac{\sigma^2}{\sigma^2 + (I_{diff}(x,y) - I_{diff}(u,v))^2} \\ d = (x-u)^2 + (y-v)^2 \\ \sigma = Internal scaling factor \end{aligned} \tag{2}$$

d is the diameter of the local surrounding of each pixel;

7.3 Merit functions

So far, all merit functions were only suitable for intramodal registration(and many more exist). A merit function that has gained wide popularity due to the fact that it can be applied to both inter- and intramodal registration problems is mutual information. In general information theory, mutual information(MI) is defined as:

$$MI = E(P(I_{Base}, I_{Match})) ln \frac{P(I_{Base}, I_{Match})}{P(I_{Base})P(I_{Match})} \tag{3}$$

where $E(P(I_{Base}, I_{Match}))$ is the expectation value of the *jointPDF*-noted as $P(I_{Base}, I_{Match})$ in Equation **??** - of the two images I_{Base} and I_{Match}, and $P(I_{Base})$ and $P(I_{Match})$ are the PDF for the gray values in the single images.

7.3 Merit functions

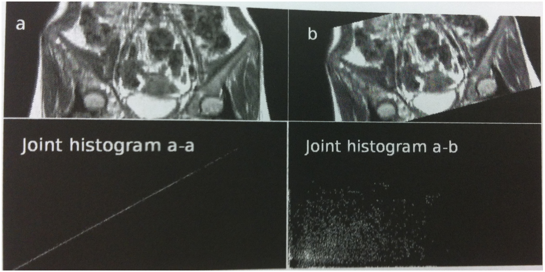


Figure 7.5: The output from the *JointHistogram$_7$.m* script. In the upper row, we see two MR images. They are the same images, but they differ by a small rotation and translation. The lower images show the output of the *JointHistogram$_7$.m* script, the joint histogram of image **a** and itself is computed(lower left). The surface plots of these joint histograms can be found in Figure **??**.

7.3 Merit functions

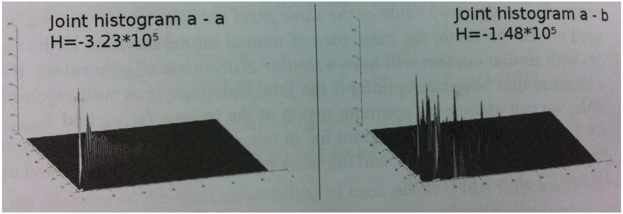


Figure 7.6: Surface plots of the joint histograms given in Figure **??**. The first histogram-which is the joint histogram of the MR image **a** from Figure **??** with itself-shows that the histograms are not scatterplots.

7.3 Merit functions

In Figure **??**, we see that there is a subtle change in the appearance of the joint histogram; since we only apply a rotation, we always have a rather large overlapping area. If we perform a translation, the number of densely populated areas does obviously change to a grater extent, with large sparsely populated areas. A script that performs such a translation for the CT- and MR-datasets named $MultimodalJointHistogramTranslation_7.m$ can be found in the $LessonData-$ folder. The joint histograms for overlap and translations in $y-$ direction of -100 and 100 pixels can be found in Figure **??**.

7.3 Merit functions

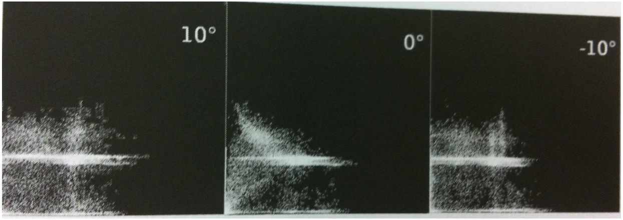


Figure 7.7: The joint histogram of two registered slices from the *LessonData* folder, *CT.jpg* and *T1.jpg*. The images were rotated against each other with an angle of -10,0, and 10; while the joint histogram for the optimum case(rotation angle 0, middle image) does not look like the lower left image in Figure **??** since the images are taken from different modalities, it is optimal in terms of disorder, as one can see when computing the mutual information as a merit function.

7.3 Merit functions

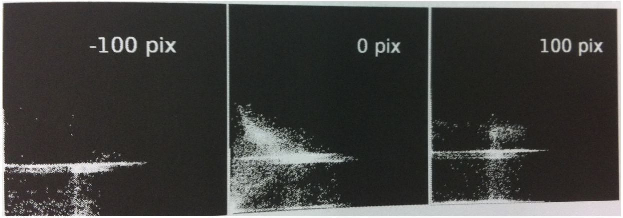


Figure 7.8: Another example for the joint histogram of two registered slices from the *LessonData* folder, *CT.jpg T1.jpg*. This time, one of the slices was translated by -100 pixels, and by 100 pixels. Since the overlap of the images is less than in the case of image rotation, the disorder in the joint histogram is increased, resulting in visibly less densely populated areas in the joint histogram.

7.3 Merit functions

We see that the a well-ordered histogram is characterized by the fact that most points in the joint histograms are accumulated in a few areas. A measure for this disorder in a PDF P, which we will call *entropy*^{2} from now on, is $Shannon'sentropy$:

$$H = -\sum_i P_i ln(P_i) \tag{4}$$

We define a merit function that utilizes mutual information as follows:

$$M_{MI} = H(I_{Base}) + H(I_{Match}) - H(I_{Base}, I_{Match}) \tag{5}$$

7.3 Merit functions

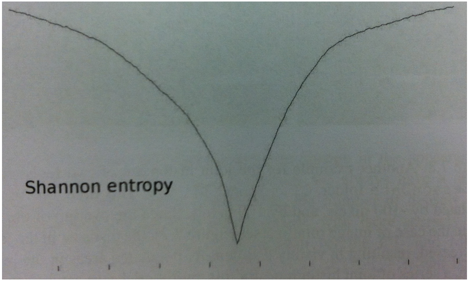


Figure 7.9: The Shannon entropy for the joint histogram as computed in *Multimo* $-$ *dalJointHistogram*$_7$*.m*. The abscissa shows the angle of relative rotation of the *T*1*.jpg* and *CT.jpg* from the *LessonData* folder, which ranges form -90 to 90. Again, an optimum is encountered when the two images match at a relative rotation of 0. The corresponding joint histogram is shown in the middle of Figure **??**

7.3 Merit functions

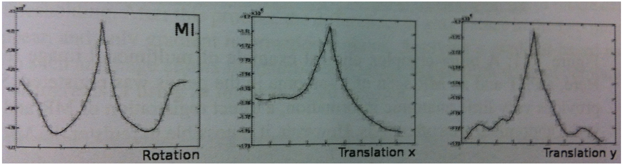


Figure 7.10: The output from the script *PlotMutualInformation2D$_7$.m*. The shape of M_{MI} is plotted when changing all three dof of rigid motion in 2D separately for the two sample images $T1$*.jpg* and CT*.jpg*. Despite me fact that this representation is ideal and that a maximum of M_{MI} can be seen, local optima are visible and may case a failure of the registration procedure.

7.3 Merit functions

For the sake of completeness, we should also add the definition of a variation of M_{MI} called *normalizedmutualinformation*:

$$M_{NMI} = \frac{H(I_{Base}) + H(I_{Match})}{H(I_{Base}, I_{Match})} \tag{6}$$

Multimodal matching always necessitates that the gray value ρ in corresponding image elements is replaced by a more general measure; Figure **??** illustrate this operations on our already well-known pig slices $CT.jpg$(the base image) and $T1.jpg$(the match image) in Figure **??**.

7.3 Merit functions

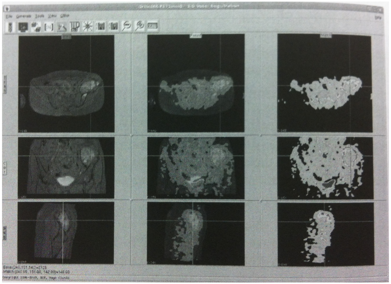


Figure 7.11: A more complex clinical example of multimodal image registration. Here, a PET and an MR scan of a sarcoma in the pelvis was registered. Since PET provides very little anatomic information, a direct registration of MR and PET data in the abdominal area often fails. However, it is possible to register the MR data with the CT data that came with the PET-scan from a PET-CT scanner using normalized mutual information.

7.3 Merit functions

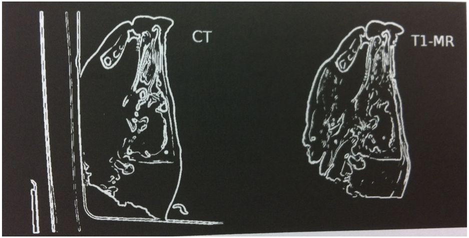


Figure 7.12: *CT.jpg* and *T1.jpg* after Sobel-filtering, optimization of image depth and thresholding. After this process, the differences in gray values in the two images disappear, and only gradient information remains. Applying a gradient-based measure to such an image allows for multi-modal image registration without computing joint histograms or using an intensity based measure.

7.3 Merit functions

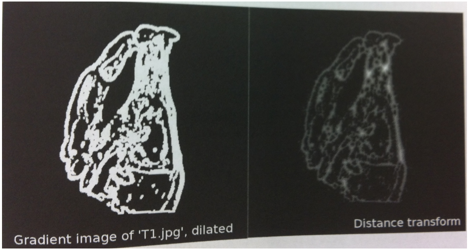


Figure 7.13: *T1.jpg* after Sobel-filtering, intensity thresholding, and dilation(left). This image can be found in the *LessonData* folder as $T1_E$*dgeImageDilated.jpg*. The right image shows its distance transform(DT_C*hamferMatch.jpg* in *LessonData*). In the chamfer-matching algorithm, the sum of the entries in the distance transform of I_{Base} at the location of non-zero pixels in I_{Match} is computed while I_{Match} moves over the base image.

7.3 Merit functions

The chamfer matching merit function M_{CM} is simply defined as:

$$\begin{aligned} M_{CM} &= \sum_{x,y} D(I_{Base}(x,y)) \\ x, y &\text{ ... } \textit{Pixels in } I_{Match} \textit{ with } \rho > 0 \\ D(I_{Base}) &\text{ ... } \textit{Distance transform of } I_{Base} \end{aligned} \tag{7}$$

Equation **??** can be understood as follows; D_{Base} - the distance transform from Section 4.3.2 - defines *grooves* in the binary gradient image I_{Base}, and if I_{Match} "drops" into those grooves, M_{CM} becomes optimal.

7.3 Merit functions

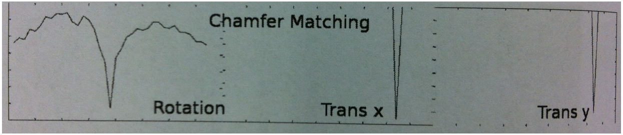


Figure 7.14: The result of *Example*7.9.4; this is a plot of the chamfer matching merit function in 3 dof through the solution of the registration problem. The plots can be directly compared to Figure **??**. As we can see, numerous local minima exist and for translation, there is no wide gradient. Therefore it is necessary to start the optimization algorithm close to the expected solution, since otherwise, finding the minimum is unlikely.

Overview

7.4 Optimization strategies

The optimization problem can be imagined as a salad bowl and marble. If we have a nice, smooth salad bowl and a nice, smooth marble, the marble can be released at any point and it will find its way to the minimum under the influence of gravity. In our less than perfect world, we may however encounter problems in optimization. Optimization algorithms can be categorized as *local* and *globalalgorithms*.

The simplex-algorithm is one of many optimization methods. other algorithms like $Powell'smethod$, $Levenberg - Marquardt$, *conjugategradient* and so on are well-documented in numerous textbooks. All this algorithms are "Local optimizers".

Globaloptimizers try to tackle the problem of finding - as the name implies-global optima.

7.4 Optimization strategies

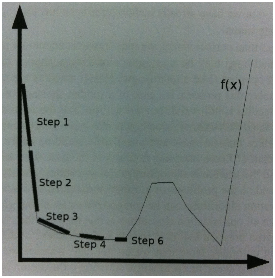


Figure 7.15: A simple illustration of the Nelder-Mead or simplx algorithm, a local optimization algorithm; a simplex in one dimension is a section of straight line that follows the gradient of a function; as the simplex approaches a minimum, $\frac{d}{dx}f(x) \simeq 0$ is true, and the simplex makes smaller steps until it collapses to a point since the gradient to right is ascending again and the algorithm terminates. It cannot find the global minimum to the right, but it is stuck in the local minimum.

Overview

7.5 Some general comments

Registration, despite the considerable complexity a full-featured implementation of a high-performance registration tool; is the more pleasant part of medical image processing compared to segmentation. So what can we do to achieve good registration results? Here is a small checklist:

- Initialization;
- Appropriate choice of the merit function;
- Appropriate choice of optimization methods;
- Avoid additional degrees-of-freedom;
- Restriction to relevant image content;

Overview

7.6 Camera calibration

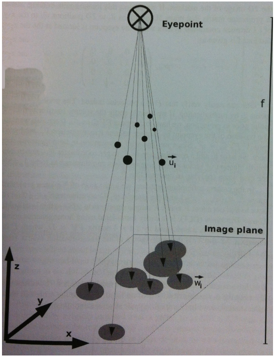


Figure 7.16: An illustration of the camera calibration problem. In this example, the imaging plane is located in the x-y plane of a Cartesian coordinate system. The normal distance of the eyepoint- the source of projection-is located at a distance **f** from the imaging plane.

7.6 Camera calibration

In order to tackle the camera calibration problem, we have to introduce another matrix, which is a projector, each matrix that fulfills the requirement $P = P^2$ is called a projection. A projection matrix that maps 3D positions $\overrightarrow{u_i}$ to 2D positions $\overrightarrow{w_i}$ to the x-y plane of a Cartesian coordinate system where the eyepoint is located at the z-axis at a distance **f** is given as:

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{f} & 1 \end{pmatrix} \tag{8}$$

7.6 Camera calibration

In the camera calibration problem, we know a set of N planar projected coordinates $\overrightarrow{w_i} \in \overrightarrow{w_0}...\overrightarrow{w_N}$ and the associated 3D coordinates $\overrightarrow{u_i} \in \overrightarrow{u_0}...\overrightarrow{u_N}$. We have to find out about V and **f**. A simple algorighm to achieve this is called the Direct Linear Transform(DLT). It is easily formulated using homogeneous coordinates. a 3D coordinate $\overrightarrow{u}$ is transformed by a 4×4 matrix $PV = D$, where P is given in Equation **??** and V is our well-known affine transformation matrix:

$$D\overrightarrow{u} = \overrightarrow{v} \tag{9}$$

7.6 Camera calibration

It is, however, necessary to renormalizes the resulting vector $\overrightarrow{u}$ by computing $\overrightarrow{w} = \frac{\overrightarrow{v}}{v_4}$, we omit the z-variable since it is of no importance for the projected screen coordinates; therefore we have a resulting pair of screen coordinates:

$$\overrightarrow{w}' = \begin{pmatrix} w_1 \\ w_2 \\ 0 \\ 1 \end{pmatrix} \tag{10}$$

7.6 Camera calibration

By expanding the matrix product and using Equation **??** we can rewrite Equation **??** as:

$$D_{11}u_1 + D_{12}u_2 + D_{13}u_3 + D_{14} = w_1 * v_4 \tag{11}$$

$$D_{21}u_1 + D_{22}u_2 + D_{23}u_3 + D_{24} = w_2 * v_4 \tag{12}$$

$$D_{41}u_1 + D_{42}u_2 + D_{43}u_3 + D_{44} = v_4 \tag{13}$$

We can insert Equation **??** in Equation **??** and **??**, which gives us two equations with twelve unknowns $D_{11}, D_{12}, D_{13}, D_{14}, D_{21}, D_{22}, D_{23}, D_{24}, D_{41}, D_{42}, D_{43}$ and D_{44}. These can be rewritten as a 12×12 matrix.

$$\begin{pmatrix} u_1 & u_2 & \cdots & -w_1 \\ 0 & 0 & \cdots & -w_2 \end{pmatrix} \begin{pmatrix} D_{11} \\ D_{12} \\ \cdots \\ D_{44} \end{pmatrix} = \overrightarrow{0} \tag{14}$$

Overview

7.7 Registration to physical space

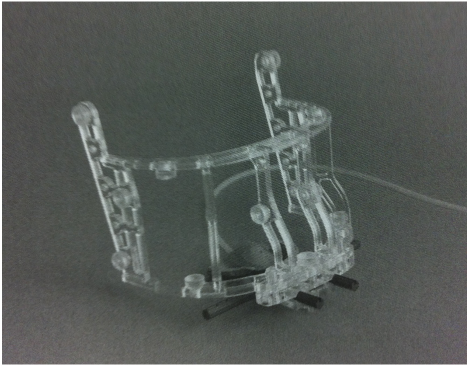


Figure 7.17: The Vogele-Bale-Hohner mouthpiece, a device for non-invasive registration using fiducial markers. The mouthpiece is personalized by means of a dental mold. A vacuum pump ensures optimum fit of the mouthpiece during imaging.

7.7 Registration to physical space

7.7.1 Rigid registration using fiducial markers and surfaces

One of the most widely used algorithms for matching ordered point sets was given by Horn. It matches two ordered point sets P_{Base} and P_{Match} by minimizing:

$$M = \frac{1}{2} \sum_{i=1}^{N} \| \overrightarrow{p} Base_i - V \overrightarrow{p} match_i \|^2 \tag{15}$$

where V is the well-known affine volume transformation matrix. After computing the centroids $\overrightarrow{\overline{p}}_{Base}$ and $\overrightarrow{\overline{p}}_{Match}$, one can define a 4×4 covariance matrix C of the point sets as:

$$C = \frac{1}{N} \sum_{i=1}^{N} (\overrightarrow{p} Match_i - \overrightarrow{\overline{p}} Match)(\overrightarrow{p} Base_i - \overrightarrow{\overline{p}} Base)^T \tag{16}$$

7.7 Registration to physical space

7.7.1 Rigid registration using fiducial markers and surfaces
Next, an anti-symmetric matrix $A = C - C^T$ is formed from C; the vector $\triangle$ is given by $\triangle = (A_{23}, A_{31}, A_{12})^T$. Using this vector, one can define a 4×4 matrix Q:

$$\mathbf{Q} = \begin{pmatrix} trC & \triangle^T \\ \triangle & C + C^T - trC\mathbf{I}_3 \end{pmatrix} \tag{17}$$

trC is the trace of matrix C, the sum of all diagonal elements. $\mathbf{I}_3$ is the 3×3 identity matrix. The translation is derived, as already mentioned before, by computing:

$$\overrightarrow{t} = \frac{1}{N} \sum_{i=1}^{N} \overrightarrow{p} Match_i - R \overrightarrow{p} Base_i \tag{18}$$

7.7 Registration to physical space

7.7.2 2D/3D registration

Registration of the patient to image data is not confined to point- or surface-based methods; it can also be achieved by comparing image data taken during or prior to an intervention with pre-interventional data.

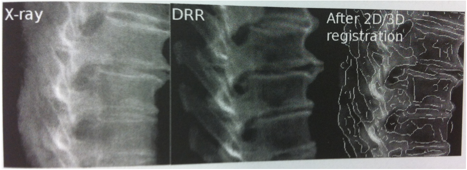


Figure 7.18: Three images of a spine reference-dataset for 2D/3D image registration. We see the reference x-ray on the left, the DRR generated from the initial guess, and the registration result. An edge image of the x-ray is overlaid over the final DRR after 2D/3D registration.

Overview

7.8 Evaluation of registration results

As usual in medical image processing, many algorithms can be defined, but they tend to be worthless if a substantial benefit for clinical application cannot be shown. Therefore, evaluation is a crucial problem here.

- Fiducial Localization error (FLE)
- Fiducial registration error(FRE)

Besides accuracy, it is also necessary to report the range of cnvergence for a measure. A report on failed registrations and on algorithm runtimes is also mandatory.

Finally, a validation should be carried out on phantom datasets which allow for determination of a ground truth such as our pig-dataset we tormented to such a large extent in this chapter.

Overview

7.9 Practical lessons

7.9.1 Registration of 2D images using cross-correlation

- Load pig head and a matrix *rotimg* is allocated for the rotated image.
- Next, we enter a loop where the rotation angle ϕ is varied from -90° to 90° in 1° steps.
- Next, we compute the correlation coefficient;
- Furthermore, the merit function becomes maximal for an optimum match.
- Finally, the contents of the merit function vector *ccvals* are displayed.

7.9 Practical lessons

7.9.1 Registration of 2D images using cross-correlation

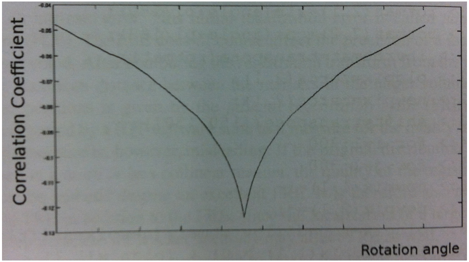


Figure 7.19: The output of this test. A registration algorithm computes the spatial transform to match two images by optimizing all dof until a minimum in the merit function is found.

7.9 Practical lessons

7.9.1 Registration of 2D images using cross-correlation

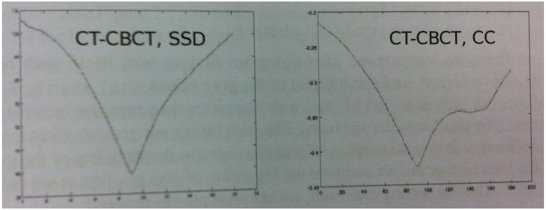


Figure 7.20: Two sample outputs from modified versions of *More2DRegistration$_7$.m*, where two registered slices of a CBCT and a CT were rotated against each other, and the result was evaluated using Equation 7.1.

7.9 Practical lessons

7.9.2 Computing joint histograms

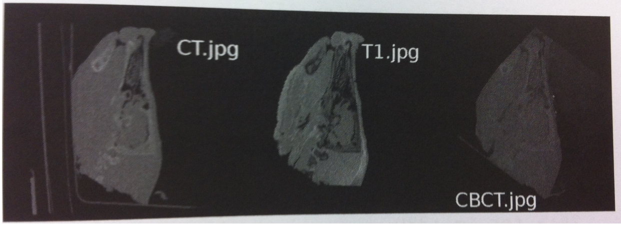


Figure 7.21: Three sample slices from co-registered volumes; *T1.jpg* is taken from an MR-volume, whereas *CT.jpg* and *CBCT.jpg* are from CT-volumes.

7.9 Practical lessons

7.9.3 Plotting the mutual information merit function

- First, a vector *mi* is defined, which holds the values for the merit function.
- Then, the histograms and joint histograms are derived.
- Next, we compute Shannon's entropy for the histograms of *img* and *rotimg*.

7.9 Practical lessons

7.9.4 Chamfer matching

For matching the thresholded gradient images of $CT.jpg$ and $T1.jpg$, we need a distance transform of I_{Base}, which is computed by the script $DTChamfermatch_7.m$.

- First, An array for the resulting chamfer matching merit function M_{CM} named *cf* is also allocated.
- Next, the image undergoes a rotation by an angle $\phi \in -180^\circ \cdots 180^\circ$ by steps of 10°.

In short words, the value of the distance transform is summed up at the location of a non-zero pixel in *rotimg*, which is our I_{Match}.

7.9 Practical lessons

7.9.5 Optimization

First, we want to explore the properties and pitfalls of numerical optimization. For this purpose, we define two very simple one-dimensional functions that can be found in Figure **??**.

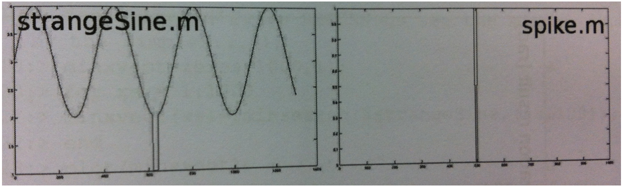


Figure 7.22: The functions defined in the scripts *StrangeSine.m* and *spike.m*. Both show somewhat unpleasant properties; the first one features a series of local minima, the other gives has a gradient of zero for most of its domain.

7.9 Practical lessons

7.9.5 Optimization

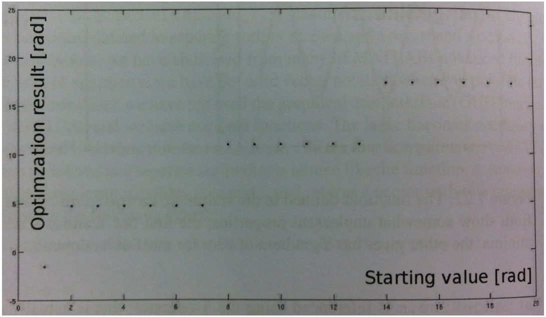


Figure 7.23: The optimization result of *strangeSine.m* in dependence of the starting value *xr*. Only for a limited range, the correct value(which is approximately 11) is found, otherwise the optimizer is struck in local minima.

7.9 Practical lessons

7.9.5 Optimization

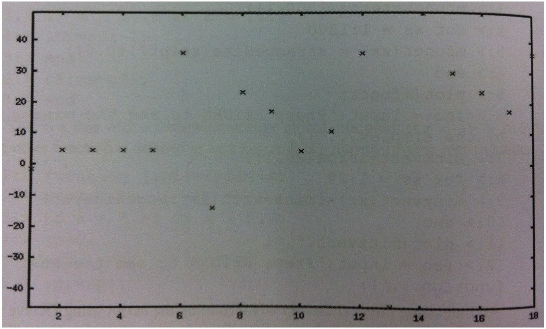


Figure 7.24: The result of optimizing *strangeSine.m* using the implementation of the simplex algorithm of Octave. It should look exactly like Figure **??**. However, the internal settings of the optimizer cause a very poor convergence behaviour.

7.9 Practical lessons

7.9.5 Optimization

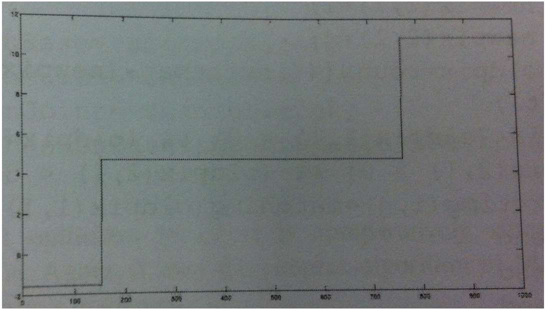


Figure 7.25: The result for the optimization of the inverse rect-function *spike.m*; the starting values *xs* range from 0 to 10 in steps of 0.01. Only if the starting value is close enough to the solution, the correct minimum(approximately 5) is found.

7.9 Practical lessons

7.9.6 The direct linear transform

- First, The projected coordinates are stored in *sevenPlanarPoints*. We don not apply any further transform to the points, therefore the projection operator is the only matrix we apply.
- Next, we construct a matrix according to Equation 7.14;

7.9 Practical lessons

7.9.7 Marker based registration

MarkerRegistration$_7$*.m* implements Eqs.7.16-7.18. The allows you to define three points $\overrightarrow{p}Base_i$ and three rotation angles px, py, pz and a translation vector *trans*.

- Lets start with the first few lines that allow you to give some input.
- Next, the point-to-point registration algorithm as given in Section 7.7.1 is implemented.
- We finny compute the eigenvectors of matrix Q from Equation 7.17;
- Finally, we compute the registration matrix by using Equation 6.19 and the translation by using Equation 7.18.

Overview

Summary

Registration is a wide field with numerous applications; as opposed to segmentation, we have an arsenal of proved algorithms at hand that can handle most clinical problems if applied properly.

We have learned about inter- and intramodal image fusion techniques, the determination of affine registration parameters from projective images, and about marker- and surface based registration.

The methods presented here only provide an overview, and for performance reasons, we have only used 2D dataset. All methods do, however, directly translate to 3D. One aspect that cannot be underestimated is the proper implementation of optimization algorithm; the optimization algorithm is as important as the merit function, and its functionality can only be determined in extensive validation series on a multitude of images.