Warping a neuro−anatomy atlas on 3D MRI data with Radial Basis Functions

Joris Korbeeck, Edwin Bennink, Arjan Jansen, Marc Koppert, Roger Lahaije, Thomas Plantenga, Bart Janssen, Bart M. ter Haar Romeny

Eindhoven University of Technology
Department of Biomedical Engineering
Biomedical Image Analysis
Eindhoven, the Netherland

■ Introduction

Neurosurgical operations require very precise navigation. Not only during resection surgery of brain tumors, also for the placement of electrodes for the electrical stimulation of some specific deep−brain structures. In particular, the placement of electrodes for stimulation of the Nucleus Subthalamicus (STN) in the deep brain may enable Parkinson patients to reduce their spontaneous tremor, which is often invalidating them.

Today, with Magnetic Resonance Imaging and Computed Tomography scanners it is possible to image the patient in three dimensions at high resolution, typically 0.5−1 mm in each direction.
However, it is often difficult to find the many individual brain nuclei on these scans due to the low contrast and the noise. The mapping of an electronic brain atlas on the scans substantially helps the navigation. The atlas is based on a mean patient, and will not fit the data. So a 3D warping method is necessary. In this paper we describe a warping based on anatomical landmark registration. The 3D deformation field is constructed by means of an interpolation based on thin plate splines, a special form of the radial basis functions. In *Mathematica* the code is very short, readible and efficient.

```
url = "http://www.bmi2.bmt.tue.
    nl/image-analysis/People/BRomeny/images/";
Show[Import[url <> "Odin.jpg"], ImageSize → 200];
```

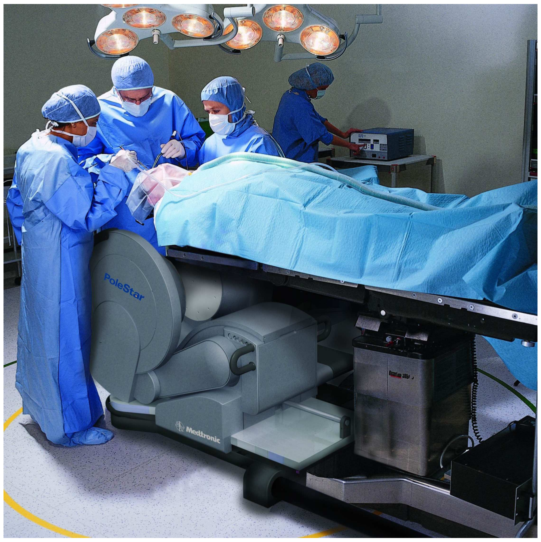


Fig. 0. Open MRI on the neurosurgical theatre enables imaging during surgery (Medtronic – Odin N20, Maastricht University Hospital, the Netherlands).

```
Show[Import[url <> "Brainatlas.jpg"], ImageSize → 400];
```

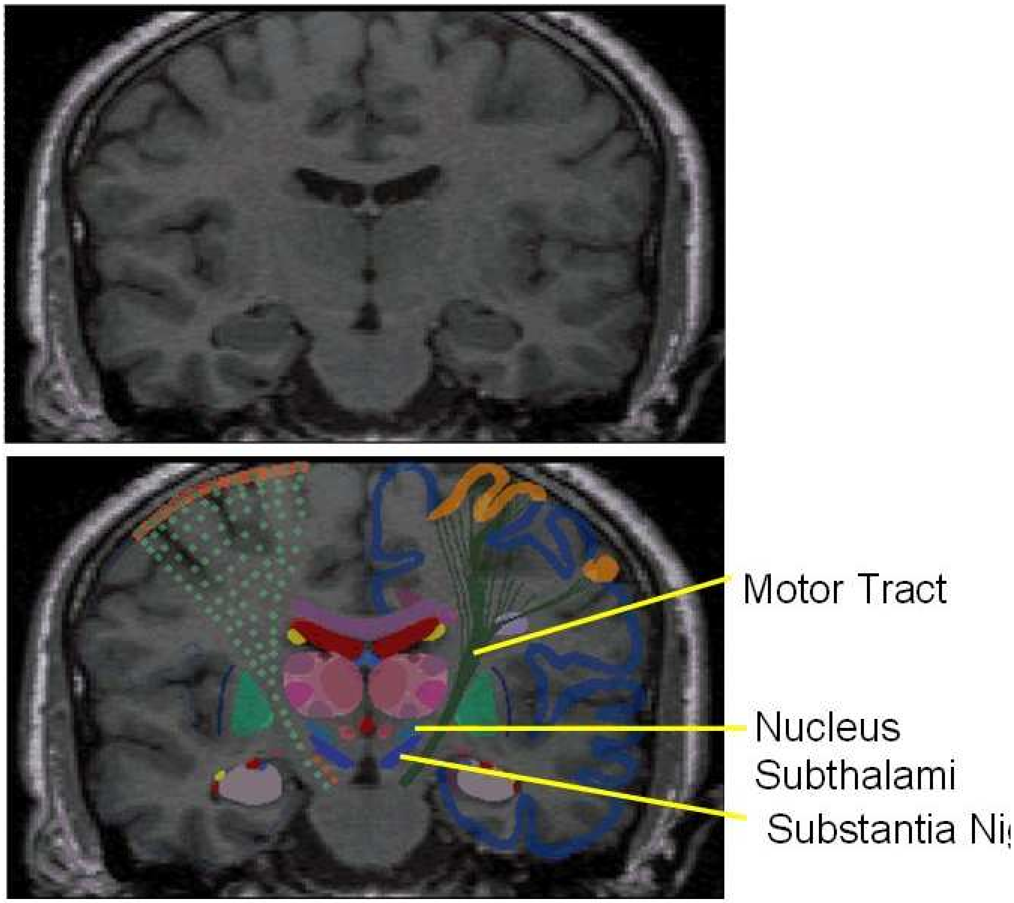


Fig. 0. Electronic brain atlas mapping on Magnetic Resonance slices.

■ Thin Plate Splines

In order to warp a brain atlas onto an MRI scan of the brain we identify so called landmarks of which the correspondence between the atlas and the scan is known. Because of this correspondence we can find the vector field that can be used for the warping procedure. The vector field is found by interpolating the know vectors at the location of the landmarks. Hence we first introduce interpolation of arbitrarily spaced points in multiple dimensions.
Imagine a thin metal plate of infinite extent that is fixed at certain points $\boldsymbol{x}_i = \{x_i, y_i\}$ at the heights f_i , i ∈ I (and neglect gravity).The metal plate has a shape such that its surface is minimally bent.The bending energy of such a plate $s(x, y)$ equals

$$(1) \qquad E_{\text{bend}}(s) = \int_{\mathbb{R}^2} \frac{\partial^2 s(x,y)}{\partial x^2} + 2\,\frac{\partial^2 s(x,y)}{\partial x\,\partial y} + \frac{\partial^2 s(x,y)}{\partial y^2}\, dx\, dy\,.$$

This is an instance of a semi−norm proposed by Jean Dychon.

In order to find the shape of a plate that is fixed at the points mentioned above we will have to find the minimizer of equation (1) such that the constraints, $s(\boldsymbol{x}_i) = f_i$ for all i ∈ I, are met. The shape can be approximated by finding that s that minimizes

$$(2) \qquad \mathrm{E(s)} = \sum_{i \in I} \left| s(\boldsymbol{x}_i) - f_i \right|^2 + \lambda\, E_{\text{bend}}(s)\,.$$

The first part of this convex energy functional makes sure the constraints are met and the second part smooths the result. $\lambda \in \mathbb{R}$ is a parameter that controlls the quality (deviation from the constraints) of the approximation. A smaller value of λ results in a better approximation, ultimately achieving interpolation when λ tends to 0.

Duchon was one of the first who recognised that the solution of the variational problem can be written in the form

$$(3) \qquad s(\boldsymbol{x}) \;=\; \sum_{i \in I} w_i \;\phi(\,\|\,\boldsymbol{x} - \boldsymbol{x}_i\,\|) \;+\; p(\boldsymbol{x})\,.$$

Here

$$\boldsymbol{\phi}[\mathbf{r\_}] = \mathbf{With}\Big[\{q = 2\}, \begin{cases} r^{2q-d}\ (\mathbf{Log}[r]) & \mathbf{EvenQ}[2\,q - d] \\ r^{2q-d} & \mathbf{OddQ}[2\,q - d] \end{cases}\Big];$$

is a so called radial basis function. We negelect q, which is a setting for the order of the norm that will be minimized. d sets the number of dimensions, for our fiducial metal plate $d = 2$. $p(\boldsymbol{x})$ is a polynomial that lies in the null space of the differential operator that appears in the bending energy. In our example it thus takes the form $a_1 \;+\; a_2\;x \;+\; a_3\;y$.

In order to find the interpolating (or approximating) function of the form of equation (3) we can simply solve a linear system of equations. Loosely said we are searching for the solution of

$$(4) \qquad \begin{pmatrix} B + \lambda\, I & Q \\ Q^{\mathsf{T}} & O \end{pmatrix} \begin{pmatrix} \boldsymbol{w} \\ \boldsymbol{a} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{pmatrix}$$

Here $\{B\}_{i,j} \;=\; \phi(\,\|\,\boldsymbol{x}_i \;-\; \boldsymbol{x}_j\,\|)$, $\{Q\}_{i,j} \;=\; \{1,\,x_i\,,\,y_i\,\}$, I is the Identity matrix and $O \;=\; \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$. Those $\boldsymbol{w}$ and $\boldsymbol{a}$ that solve the linear system of equation (4) can be plugged into equation (3) in order to find the expression for the shape we are looking for.

The method presented above is very elegant and educative. However, it is not the most efficient way to perform thin plate spline interpolation on non–equispaced grids in the sense of computational complexity. A fast variational multi–level method is presented by Arigovindan et. al. Their method is orders of magnitude faster than the scheme presented here.

□ Implementation

We implemented the method that is sketeched above for a 2D metal plate for N Dimensions. The module takes takes input similar to the built–in Inerpolation function and returns a compiled function. The approximation accuracy is set to a fixed value of $\lambda = 10^{-6}$.

Input:

data = {
$\{\{x_1\,,\,y_1\,,\ \ldots\},\,\{f_1\,\}\},$
$\{\{x_2\,,\,y_2\,,\ \ldots\}.\{f_2\,\}\},$
…,
$\{\{x_n\,,\,y_n\,,\ \ldots\},\,\{f_n\,\}\}$
}

Output:

A compiled function
f[x,y,...]→ℝ

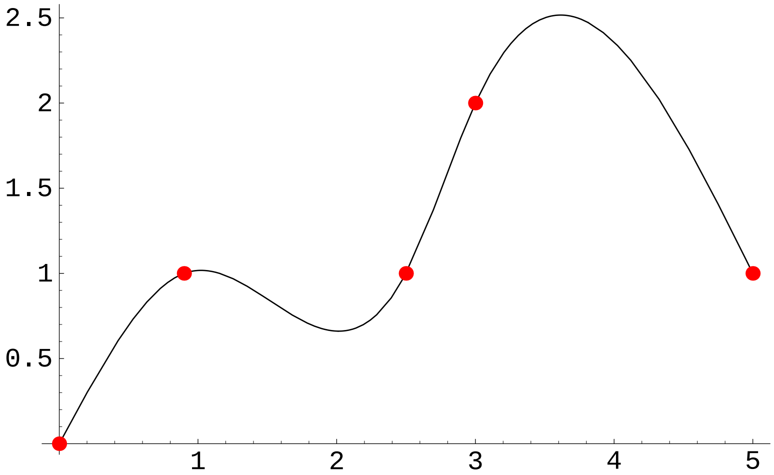
```
In[50]:= If[$VersionNumber < 6.,
           << LinearAlgebra`
         ];
         $HistoryLength = 0;
```

```
Off[LinearSolve::"luc"];
Off[General::"spell1"];
RBFInterpolation[data_] :=
  Module[{M = Length[data], d = Length[data⟦1, 1⟧], φ, λ = 10^-6 ,
      f = Flatten[N[data⟦All, 2⟧]], points = N[data⟦All, 1⟧], BMatrix,
      QMatrix, OMatrix, A, b, w, a, function, p, retval, rules, fallback},
    (∗ Definition of  the RBF as proposed by Duchon ∗)
    φ[r_] = With[{q = 2}, ⎧ r^(2 q−d) (Log[r])    EvenQ[2 q − d]  ];
                          ⎩ r^(2 q−d)             OddQ[2 q − d]
     (∗ Help mathematica to resolve this limit. ∗)
    φ[0.] = 0.;

(∗ Construct the matrix that should be inverted ∗)
    OMatrix = Table[0., {d + 1}, {d + 1}];
BMatrix = Map[
        φ[Norm[#]] &,
        Outer[Plus, points, −points, 1],
        {2}] ;
QMatrix = Map[Prepend[#, 1] &, points];

    (∗ Version 6 wants ArrayFlatten instead of BlockMatrix ∗)
    A = If[$VersionNumber ≥ 6.,
ArrayFlatten[( BMatrix + λ IdentityMatrix[M]   QMatrix )],
             ( QMatrixᵀ                         OMatrix )
        BlockMatrix[( BMatrix + λ IdentityMatrix[M]   QMatrix )]
                    ( QMatrixᵀ                         OMatrix )
      ];
b = Flatten[Append[f, Table[0., {d + 1}]]];
(∗ Solve the linear system of equations we just constructed ∗)
    {w, a} = Block[{x, w, a},
        x = LinearSolve[A, b]; (∗Chop[PseudoInverse[A].b]∗)
        w = Take[x, {1, −(d + 2)}];
        a = Take[x, −(d + 1)];
        {w, a}
      ];

    (∗ some trickery to get the function to compile correctly.  ∗)
    p = Symbol /@ Table["p" <> ToString[i], {i, d}];
    Φ = Map[φ(||p − #||) &, points];

    function = {
        Table[{p⟦i⟧, _Real}, {i, d}],
        Evaluate[ a.Prepend[p, 1] + w.Φ]
      };
    (∗dirty.... 0 log 0 evaluates as Indeterminate. A
        diry way around this is by looking just next to it ∗)
    rules = Table[p⟦i⟧ → p⟦i⟧ + $MachineEpsilon, {i, d}];
    fallback = {
        Table[{p⟦i⟧, _Real}, {i, d}],
        Evaluate[ a.Prepend[p, 1] + w.Φ /. rules]
      };

    retval = Compile @@ function;
    retval⟦−1⟧ = Compile @@ fallback;
    retval

  ]
```

Example 1D

We first show an example of interpolation through non equi–spaced points. Notice this can also be achieved using the built–in Interpolation functionality. The curve is smooth and indeed passes through the points it should interpolate.

```
data = {{{0}, {0}}, {{0.9}, {1}},
   {{2.5}, {1}}, {{3}, {2}}, {{5}, {1}}};
f = RBFInterpolation[data];
Plot[
  f[x],
  {x, 0, 5}, Epilog → {PointSize[0.02],
    Hue[1], Point /@ Partition[Flatten[data], 2]}];
```



Example 2D

Some image compression methods employ interpolation as a reconstruction method. Values of "important" pixels are usually stored in a tree structure in order to reduce overhead. As an example we randomly select approximately a little bit less than 14% of the pixels of an image of Lena:

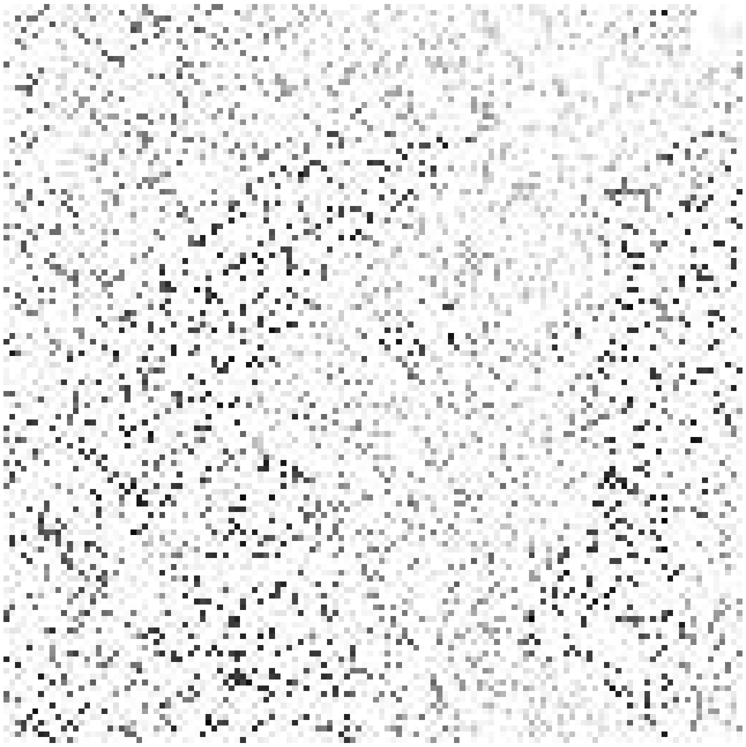
```
In[1]:= image = Import["~/Data/lena128x128.bmp"]〚1, 1〛 // N;
        {m, n} = Dimensions[image];
```

```
In[41]:= α = 0.15;
        samplepositions = Extract[
          Flatten[Outer[List, Range[n], Range[m]], 1],
          Union[Sort[
            Table[
             {Random[Integer, m n]}
             , {Round[α m n]}
            ]
           ]]
         ];
        values = Transpose[{Extract[image, samplepositions]}];
        data = N[MapThread[List, {samplepositions + 0.2, values}]];
                        Length[values]
        Print["Using ", ────────────── 100.,
                             m n
         "% of the pixels of the original image."];
```

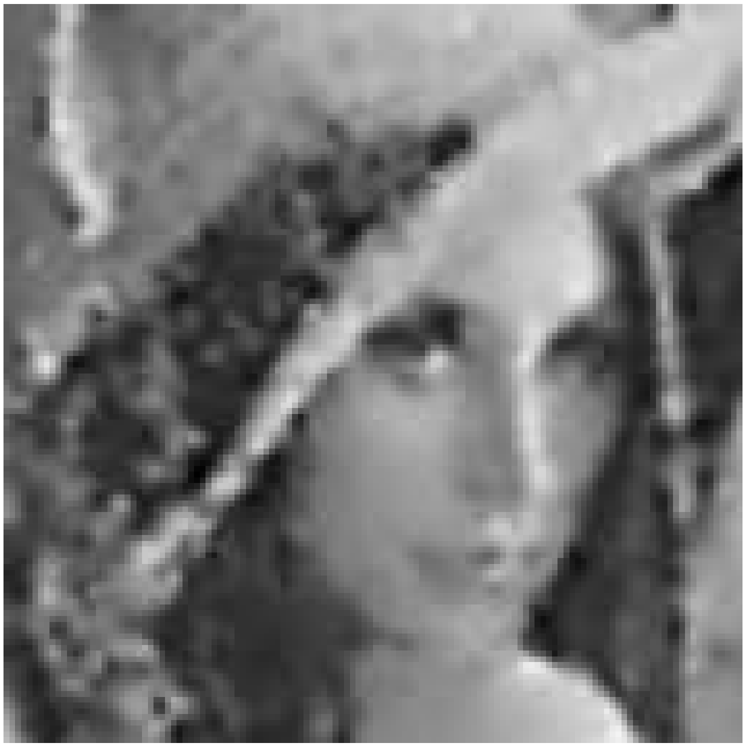
```
Using 13.9221% of the pixels of the original image.
```

A visualisation of the selected samples is shown below.

```
In[55]:= SetOptions[ListDensityPlot, Mesh → False,
          PlotRange → {0, 254}, Frame → False];
         ListDensityPlot[
          SparseArray[
           samplepositions → Flatten[values], Dimensions[image], 254
          ]
         ];
```



```
In[57]:= interpolatedimage = RBFInterpolation[data];
         ListDensityPlot[
          Table[
           interpolatedimage[x, y],
           {x, 1, 128}, {y, 1, 128}
          ]
         ];
```



The reconstructed image clearly shows the face of lena. The edges are not very sharp, which results in poor perceptual quality. Off course one can not expect sharp edges when interpolating random samples with the smoothest function possible. More sophisticated interpolation methods (employing, for example, anisotropic non–linear diffusion) or smarter sample selection significantly increase the quality of the reconstructed image at low compression rates.

□ BrainWeb Reference Points

□ Talairach Reference Points

■ References:

1. Bookstein, F. (1989) Principal warps: thin−plate splines and the decomposition of deformations. IEEE Trans. Pattern Analysis Machine Intelligence 11(6): 567−585.

2. Carr J.C., Fright W.R., Beatson R.K. (1997) Surface interpolation with radial basis functions for medical imaging. IEEE transactions on medical imaging, Vol. 16, No. 1.

3. Free S.L., O'Higgins P, Maudgil D.D., Dryden I.L., (2001) Landmark−Based morphometrics of the normal adult brain using MRI. NeuroImage 13 801−813.

4. Grachev I.D., Berdichevsky D., Rauch S.L. (1999) A method for assessing the accuracy of intersubject registration of the human brain using anatomic landmarks. NeuroImage 9, 250−268.

5. Levin D., Dey D., Slomka P.J. (2004) Acceleration of 3D, nonlinear warping using standard video graphics hardware: implementation and initial validation. Computerized Medical Imaging and Graphics 28: 471−483.

6. Nowinski W.L., Thirunavuukarasuu A. (2001) Atlas−assisted localization analysis of functional images. Medical Image Analysis 5, 207−220.

7. Nowinski W.L., (2001) Technical Report: Modified Talairach Landmarks. Acta Neurochir 143: 1045−1057 Wien.

8. Nowinski W.L., Belov D. (2003) The Cerefy Neuroradiology Atlas: a Talairach−Tournoux atlas−based tool for analysis of neuroimages available over the internet. NeuroImage 20, 50−57.

9. Rizzo G., Scifo P., Gilardi M.C. (1997) Matching a computerized brain atlas to multimodal medical images, NeuroImage 6, 59−69.

10. Talairach J., Tournoux P. (1988) Co−planar stereotactic atlas of the human brain. Georg Thiem Verlag/Thieme Medical Publishers, Stuttgart New York.

11. Talairach J., Tournoux P. (1993) Referentially oriented cerebral MRI anatomy. Atlas of stereotaxic anatomical correlations for gray and white matter. GeorgThieme Verlag/Thieme Medical Publishers, Stuttgart New York.

12. Verard L., Allain P., Travere J.M. (1997) Fully automatic identification of AC and PC landmarks on brain MRI using scene analysis. IEEE transactions on medical imaging, Vol. 16, No. 5.

13. M.Arigovindan,M.Sühling,P.Hunziker,and M.Unser.Variational image reconstruction from arbitrarily spaced samples:Variational image reconstruction from arbitrarily spaced samples:A fast multiresolution spline solution.IEEE−TIP,14(4):450-460,April 2005.

14. J. Duchon, in Constructive theory of functions of several variables,W. Schempp and K. Zeller, editors. *Splines minimizing rotation−invariant semi−norms in Sobolev spaces*. Springer Verlag, 1976.

15. J. Kybic, T. Blu, and M. Unser. Generalized sampling: a variational approach - part I: Theory. *IEEE Transactions on Signal Processing*, 50:1965-1976, August 2002.

16. URL: http://www.bmi2.bmt.tue.nl/image−analysis/People/BRomeny/azm/index.htm