Pixelwise image registration methods

Jan Kybic

2020-2023

Unser: Splines. A perfect fit for signal and image processing. 1999

Polynomial splines

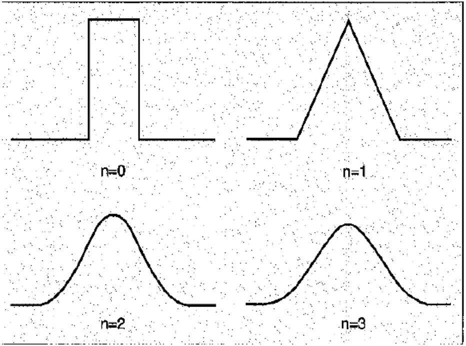
Splines are piecewise polynomials with pieces that are smoothly connected together. The joining points of the polynomials are called *knots*. For a spline of degree n, each segment is a polynomial of degree n, which would suggest that we need $n+1$ coefficients to describe each piece. However, there is an additional smoothness constraint that imposes the continuity of the spline and its derivatives up to order (n-1) at the knots, so that, effectively, there is only one degree of freedom per segment. Here, we will only consider splines with uniform knots and unit spacing. The remarkable result, due to Schoenberg [70],

$$s(x) = \sum_{k \in Z} c(k) \beta^n (x - k),$$

B-splines

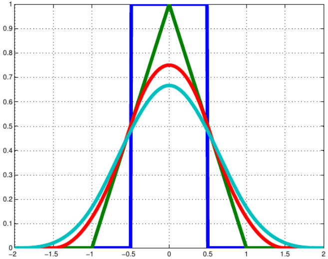
$$\beta^0(x) = \begin{cases} 1, & -\frac{1}{2} < x < \frac{1}{2} \\ \frac{1}{2}, & |x| = \frac{1}{2} \\ 0, & \text{otherwise} \end{cases}$$

$$\beta^n(x) = \underbrace{\beta^0 * \beta^0 * \cdots * \beta^0(x)}_{(n+1)\text{ times}}.$$



B-splines

Haar	β_0
linear	β_1
quadratic	β_2
cubic	β_3

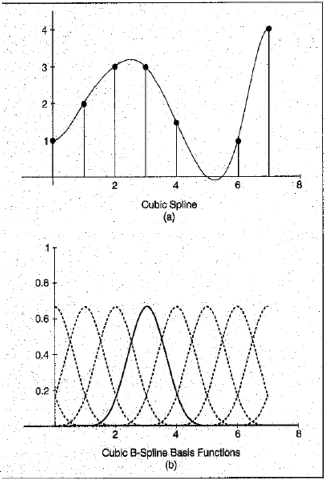


- Basis for splines: $f(x) = \sum_i c_i \, \beta(x - i)$
- Generation: $\beta_{n+1} = \beta_n * \beta_0$

Cubic B-spline

$$\beta^3(x) = \begin{cases} \frac{2}{3} - |x|^2 + \frac{|x|^3}{2}, & 0 \le |x| < 1 \\ \frac{(2-|x|)^3}{6}, & 1 \le |x| < 2 \\ 0, & 2 \le |x|, \end{cases}$$

Signal interpolation



Now, given the signal samples $s(k)$, we want to determine the coefficients $c(k)$ of the B-spline model (1) such that we have a perfect fit at the integers; i.e., $\forall k \in Z$,

$$\sum_{l \in Z} c(l) \beta^{n}(x-l) \Big|_{x=k} = s(k).$$

Using the discrete B-splines, this constraint can be rewritten in the form of a convolution

$$s(k) = (b_1^n * c)(k). \tag{12}$$

Finding B-spline coefficients

$$s(k) = (b_1^n * c)(k). \tag{12}$$

Defining the inverse convolution operator

$$(b_1^n)^{-1}(k) \quad \overset{z}{\leftrightarrow} \quad 1/B_1^n(z),$$

the solution is found by inverse filtering (cf. [97])

$$c(k) = (b_1^n)^{-1} * s(k). \tag{13}$$

Since b_1^n is symmetric FIR (finite impulse response), the so-called direct B-spline filter $(b_1^n)^{-1}$ is an all-pole system that can be implemented very efficiently using a cascade of first-order causal and anti-causal recursive filters [93], [96]. This algorithm is stable numerically and is faster and easier to implement than any other numerical technique.

IIR filter

Box 2. Fast Cubic Spline Interpolation

By sampling the cubic B-spline (6) at the integers, we find that

$$B_1^3(z) = (z + 4 + z^{-1})/6.$$

Thus, the filter to implement is

$$(b_1^3)^{-1}(k) \overset{z}{\leftrightarrow} \frac{6}{z+4+z^{-1}} = 6\left(\frac{1}{1-z_1 z^{-1}}\right)\left(\frac{-z_1}{1-z_1 z}\right)$$

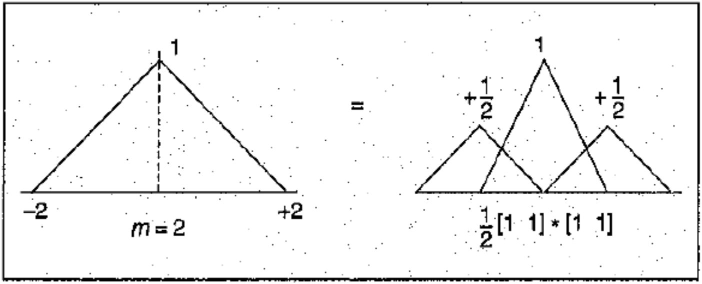
,

with $z_1 = -2 + \sqrt{3}$. Given the input signal values $\{s(k)\}_{k=0,\ldots,N-1}$ and defining $c^-(k) = c(k)/6$, the right-hand-side factorization leads to the following recursive algorithm

$$c^+(k) = s(k) + z_1 c^+(k-1), \quad (k = 1, \ldots, N-1)$$

$$c^-(k) = z_1\left(c^-(k+1) - c^+(k)\right), \quad (k = N-2, \ldots, 0),$$

Two-scale relation



where $h_m^0(k)$ is the filter whose z-transform is $H_m^0(z) = \sum_{k=0}^{m-1} z^{-k}$ (discrete pulse of size m). By convolving this equation with itself $(n+1)$-times and performing the appropriate normalization, one finds that

$$\varphi^n(x/m) = \sum_{k \in Z} h_m^n(k) \varphi^n(x-k), \qquad (29)$$

where

$$H_m^n(z) = \frac{1}{m^n}\left(H_m^0(z)\right)^{n+1} = \frac{1}{m^n}\left(\sum_{m=0}^{m-1} z^{-k}\right)^{n+1}. \qquad (30)$$

Spline pyramid

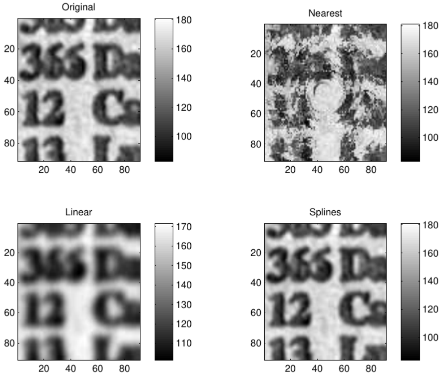
Let $P_{2^i} s = s_i$ denote the minimum error approximation of some continuously defined signal $s(x) \in L_2$ at the scale $m = 2^i$. We choose to represent it by the following expansion

$$P_{2^i} s = \sum_{k \in Z} c_{2^i}(k) \varphi(x / 2^i - k), \tag{31}$$

$$c_{2^i}(k) = (\overset{\circ}{b} * c_{2^{i-1}})(2k).$$

Image interpolation

36 rotations by 10°.



Resulting images (zoom).

Splines - summary

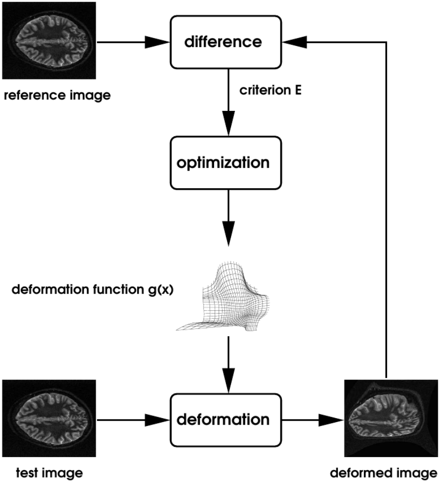
- To represent a continuous function in a discrete basis
- Compact support
- Simple to evaluate (low-order polynomials)
- Smooth, continuously differentiable (up to some order)
- Good approximation properties
- Uniform B- splines
  - coefficients fast to calculate
  - multiscale version fast to calculate and exact
- Applications: interpolation, approximation, signal/image transformations, multiscale processing

Thevenaz: Optimization of Mutual Information for Multiresolution Image Registration, IEEE TMI 2000

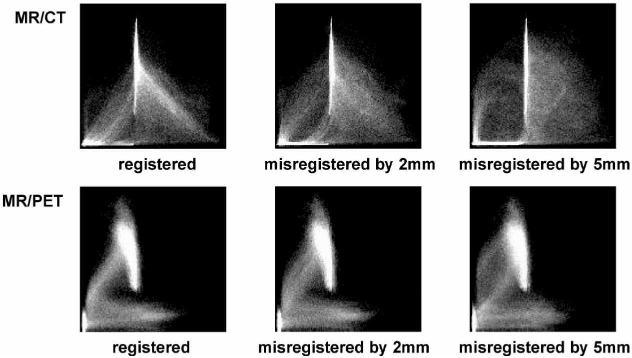
Key points

- Registration by optimization
- a pixel-based similarity criterion - mutual information
- B-spline representation
- multiresolution

Registration as minimization



Joint intensity histograms



Mutual Information

$I = H(X) + H(Y) - H(X,Y) \geq 0$, entropy $H(X) = -\sum_i P(x_i) \log P(x_i)$,
$H(X,Y) = \sum_i \sum_j P(X_{i,} Y_j) \log P(X_i, Y_j)$
Note that $\lim_{p \to 0^+} p \log p = 0$
Negative MI:

$$S(\boldsymbol{\mu}) = -\sum_{\iota \in L_T} \sum_{\kappa \in L_R} p(\iota, \kappa; \boldsymbol{\mu}) \cdot \log_2 \left(\frac{p(\iota, \kappa; \boldsymbol{\mu})}{p_T(\iota; \boldsymbol{\mu}) p_R(\kappa; \boldsymbol{\mu})} \right).$$

Histogram estimation

Parzen window

$$\tilde{p}_N(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{w((x - x_i)/\varepsilon(N))}{\varepsilon(N)}$$

$$\begin{aligned} h(\iota, \kappa; \boldsymbol{\mu}) = \frac{1}{\varepsilon_T \varepsilon_R} \sum_{\mathbf{x}_i \in V} & w(\iota/\varepsilon_T - f_T(\mathbf{g}(\mathbf{x}_i; \boldsymbol{\mu}))/\varepsilon_T) \\ & \cdot w(\kappa/\varepsilon_R - f_R(\mathbf{x}_i)/\varepsilon_R) \end{aligned}$$

Criterion model

Deformation parameters μ

Taylor expansion

$$S(\boldsymbol{\mu}) = S(\boldsymbol{\nu}) + \sum_i \frac{\partial S(\boldsymbol{\nu})}{\partial \mu_i} (\mu_i - \nu_i) + \frac{1}{2} \sum_{i,j} \frac{\partial^2 S(\boldsymbol{\nu})}{\partial \mu_i} \partial \mu_j (\mu_i - \nu_i)(\mu_j - \nu_j) + \cdots.$$

the gradient $\boldsymbol{\nabla} S$ as

$$\boldsymbol{\nabla} S = \left[\frac{\partial S}{\partial \mu_1}, \frac{\partial S}{\partial \mu_2}, \cdots \right].$$

Hessian approximation

$$
\begin{aligned}
&\frac{\partial^2 S}{\partial \mu_1\, \partial \mu_2} \\
&\quad \approx \frac{1}{\log_e(2)} \left(\sum_{\iota \in L_T} \frac{\partial p_T(\iota)}{\partial \mu_1} \frac{\partial p_T(\iota)}{\partial \mu_2} \frac{1}{p_T(\iota)} \right) \\
&\qquad - \frac{1}{\log_e(2)} \left(\sum_{\iota \in L_T} \sum_{\kappa \in L_R} \frac{\partial p(\iota, \kappa)}{\partial \mu_1} \frac{\partial p(\iota, \kappa)}{\partial \mu_2} \frac{1}{p(\iota, \kappa)} \right).
\end{aligned}
$$

- from first-order derivatives

Standard optimizers

The steepest-gradient descent is a minimization algorithm that can be succinctly described by

$$\boldsymbol{\mu}^{(k+1)} = \boldsymbol{\mu}^{(k)} - \Gamma \boldsymbol{\nabla} S\left(\boldsymbol{\mu}^{(k)}\right). \tag{32}$$

Its local convergence is guaranteed, although it may be very slow. A key problem is the determination of the appropriate scaling diagonal matrix Γ.

The Newton method can be described by

$$\boldsymbol{\mu}^{(k+1)} = \boldsymbol{\mu}^{(k)} - \left(\nabla^2 S\left(\boldsymbol{\mu}^{(k)}\right)\right)^{-1} \boldsymbol{\nabla} S\left(\boldsymbol{\mu}^{(k)}\right). \tag{33}$$

Marquardt-Levenberg

$$[\mathcal{H}S(\boldsymbol{\mu})]_{i,j} = [\nabla^2 S(\boldsymbol{\mu})]_{i,j}\,(1 + \delta_{i,j}\,\lambda)$$

$$\boldsymbol{\mu}^{(k+1)} = \boldsymbol{\mu}^{(k)} - \left(\mathcal{H}S\left(\boldsymbol{\mu}^{(k)}\right)\right)^{-1} \boldsymbol{\nabla} S\left(\boldsymbol{\mu}^{(k)}\right)$$

adaptive λ

Multiresolution

32×32

64×64

128×128

256×256

Kybic, Unser: Fast Parametric Elastic Image Registration. 2003

Key points

- Pixelwise similarity criterion
- Elastic (nonlinear) registration
- B-spline representation of the transformation and image

Cost function

$$
\begin{aligned}
E &= \frac{1}{||I||} \sum_{\mathbf{i} \in I} e_{\mathbf{i}}^2 = \frac{1}{||I||} \sum_{\mathbf{i} \in I} (f_w(\mathbf{i}) - f_r(\mathbf{i}))^2 \\
&= \frac{1}{||I||} \sum_{\mathbf{i} \in I} (f_t^c(\mathbf{g}(\mathbf{i})) - f_r(\mathbf{i}))^2
\end{aligned}
$$

Image interpolation

using uniform B-splines:[3]

$$f_t^c(\mathbf{x}) = \sum_{\mathbf{i} \in I_b \subset \mathbb{Z}^N} b_{\mathbf{i}} \beta_n(\mathbf{x} - \mathbf{i}) \qquad (2)$$

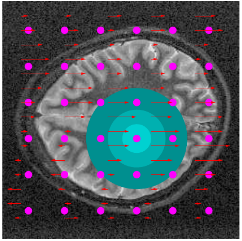
where β_n is a tensor product of B-splines of degree n, that is $\beta_n(\mathbf{x}) = \prod_{k=1}^{N} \beta_n(x_k)$, with $\mathbf{x} = (x_1, \ldots, x_N)$.

Deformation model

$$\mathbf{g}(\mathbf{x}) = \mathbf{x} + \sum_{\mathbf{j} \in J} \mathbf{c}_{\mathbf{j}} \varphi_{\mathbf{j}}(\mathbf{x})$$

$$\mathbf{g}(\mathbf{x}) = \mathbf{x} + \sum_{\mathbf{j} \in I_c \subset \mathbb{Z}^N} \mathbf{c}_{\mathbf{j}} \beta_{n_m}(\mathbf{x}/\mathbf{h} - \mathbf{j})$$

Spline based deformation



- Approximation properties $\rightarrow$ precision
- Short support $\rightarrow$ speed
- Scalability
- Representability of linear transforms

$$\mathbf{g}(\mathbf{x}) = \mathbf{x} + \sum_{\mathbf{i}\in\mathbb{Z}^2} \mathbf{c}(\mathbf{i})\,\beta(\mathbf{x}/\mathbf{h} + \mathbf{d} - \mathbf{i})$$

Optimization

If the step is successful, then the proposed point is accepted, $\mathbf{c}^{(i+1)} = \mathbf{c}^{(i)} + \Delta\mathbf{c}^{(i)}$. Otherwise, a more conservative update $\Delta\mathbf{c}^{(i)}$ is calculated, and the test is repeated.

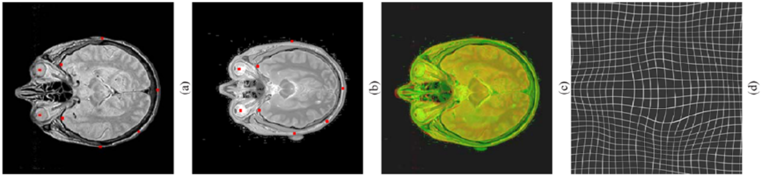
1) *Gradient descent with feedback step size adjustment* with update rule: $\Delta\mathbf{c}^{(i)} = -\mu\nabla_{\mathbf{c}}E(\mathbf{c}^{(i)})$. After a successful step, μ is multiplied by μ_f, otherwise it is divided by μ'_f.[5]
2) *Gradient descent with quadratic step size estimation*. We choose a step size μ^* minimizing the following approximation of the criterion around $\mathbf{c}^{(i)}$: $E(\mathbf{c}^{(i)}+\mathbf{x}) = E(\mathbf{c}^{(i)}) + \mathbf{x}^T\nabla_{\mathbf{c}}E(\mathbf{c}^{(i)}) + \alpha||\mathbf{x}||^2$, where α is identified from the two last calculated criterion values E. As a fallback strategy, the previous step size is divided by μ'_f, as above.

Landmarks

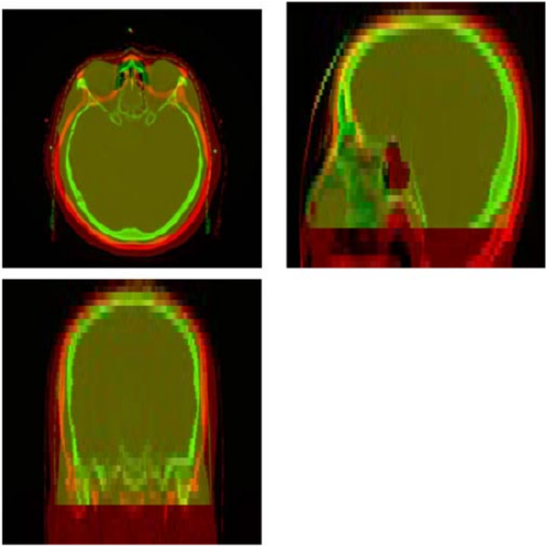
of corresponding points together. We augment the data part of the criterion E with a term E_s, corresponding to the potential energy of the springs, and minimize the sum of the two: $E_c = E + E_s$. The spring term is

$$E_{\mathbf{s}} = \sum_{i=1}^{S} \alpha_i \left|\left|\mathbf{g}(\mathbf{x}_i) - \mathbf{z}_i\right|\right|^2 \qquad (5)$$

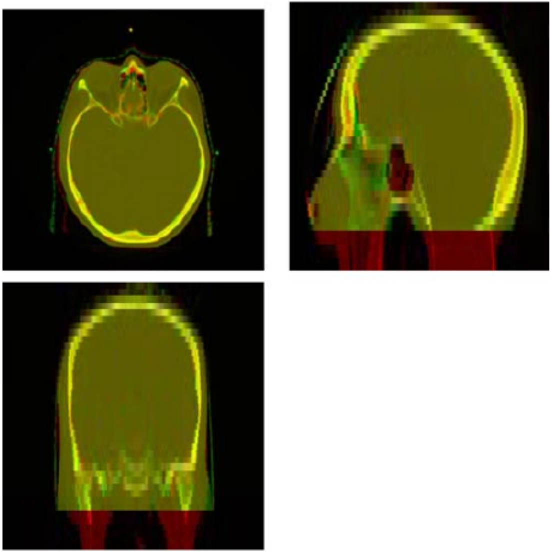
Examples



Examples (2)



Examples (3)

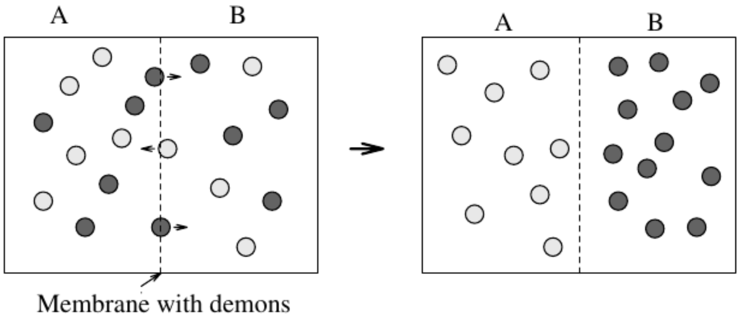


Thirion: Image matching...analogy with Maxwell's demons. MIA 1998

Key points

- Non-rigid, no global deformation model
- Registration by boundary motion
- Needs to find boundaries / keypoints
- Iterative
- Localize points/bounaries and interpolate
- Fast, local and parallelizable

Maxwell demons



Second law of thermodynamics: The total entropy of an isolated system never decreases. Isolated systems evolves towards thermodynamic equilibrium (maximum entropy).

Deformable model

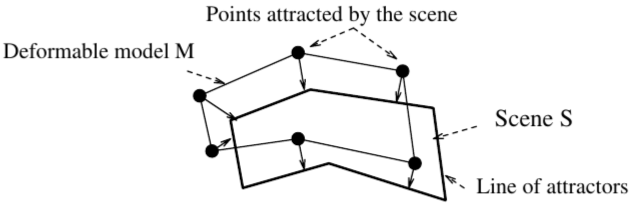


Figure 2. Deformable model with attraction.

.

$$\vec{f}(P) = \sum_{P' \in S} \frac{K(P, P')}{D(P, P')} \vec{PP'}.$$

K similarity, D distance, complexity $O(N^2)$, similar to ICP and CPD

Diffusion model

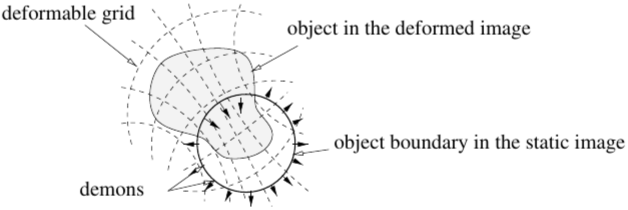
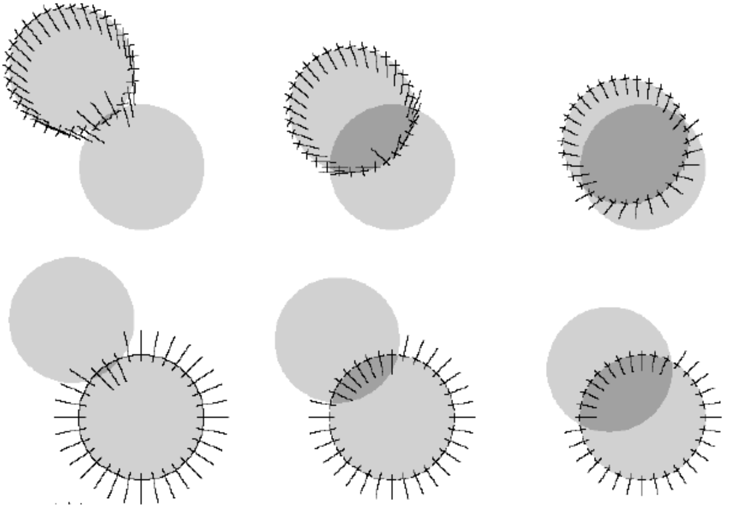


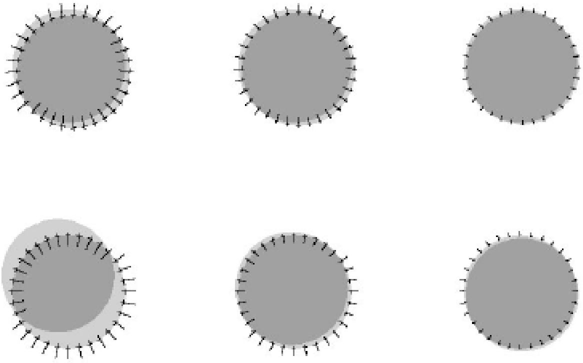
Figure 1. Diffusing models: a deformed image, considered as a deformable grid, is diffusing through the contours of the objects in the static image, by the action of effectors, called demons, situated in these interfaces.

Demons for registration



- demons on boundary →normal forces. Top: Closest point attraction. Bottom: *dark* → inside, white → outside.

Demons for registration (2)



next iterations

Initialization

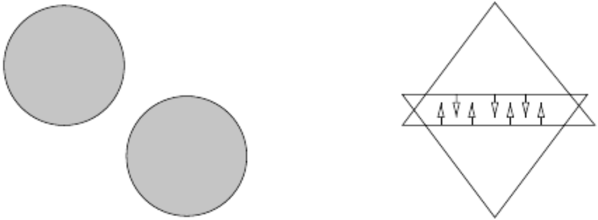


Figure 7. Example of problematic initializations: left, when the two objects to be matched do not overlap, diffusing models are inefficient. Right, with an attraction model that does not take polarity into account, and with forces decreasing with the distance, the model can get trapped in a local minimum.

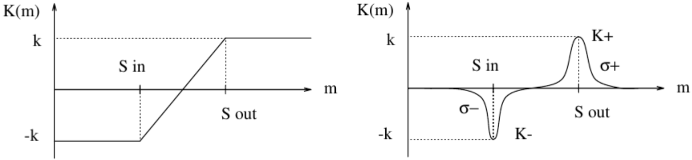
Algorithm

1. extract demons from image S, find normals (gradients)
2. Until convergence
   2.1 Compute elementary demon forces
   2.2 *Fit* global update model or *smooth* by a Gaussian
   2.3 Update transformation

Variants

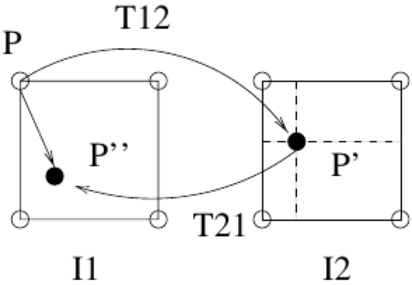
1. Complete grid demons, force from optical flow, multiscale
2. Contour demons on edges, rigid/nonrigid transformation by least squares fitting + outlier rejection,

$$\vec{f}(P) = K_{s_{\text{in}}, s_{\text{out}}}(m(P'))\vec{n}$$



3. Demons for segmented images, on boundaries, rigid/nonrigid transformation, different classes → constant magnitude forces

Bijectivity



3D example

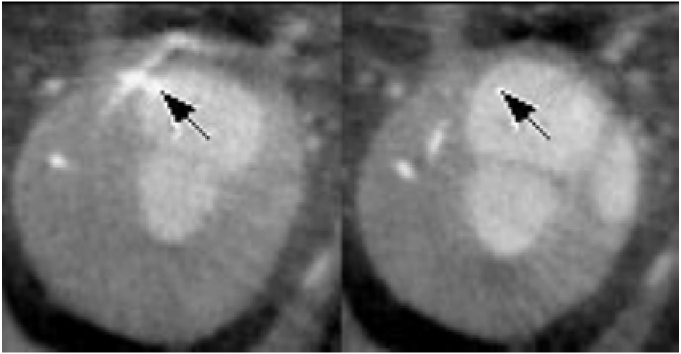


Figure 18. Corresponding diastolic and systolic slice before matching (dog).

3D examples

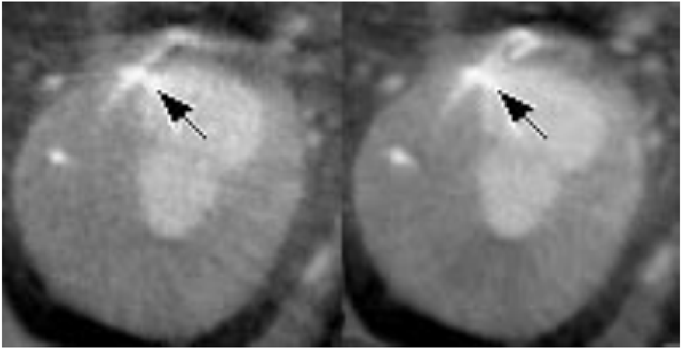


Figure 19. Corresponding diastolic and systolic slices after 3-D matching and re-sampling.

Segmentation example

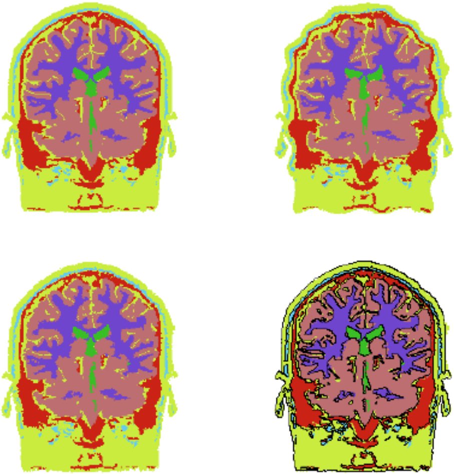


Figure 17. Top left, original label image (I_1); top right, deformed label image (I_2). Bottom left, I_2 deformed toward I_1 using the implementation 'demons 3'; bottom right, deformed I_2 with a superimposition of I_1 contours.

Intersubject registration example

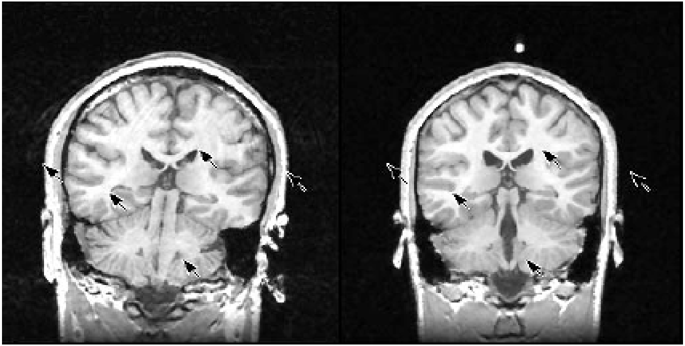


Figure 24. Two slices of two different patients ($256 \times 256 \times 128$ voxels).

Intersubject registration example (2)

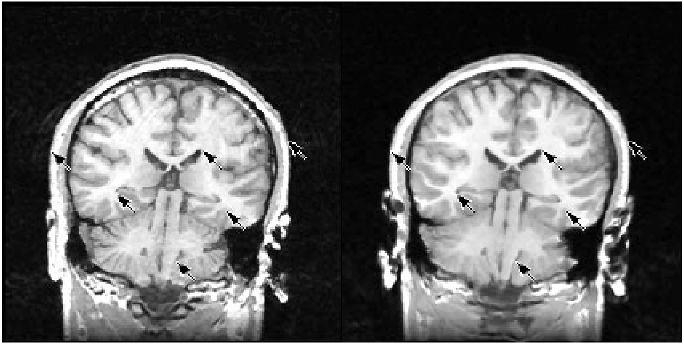


Figure 25. The two different patients after automatic matching.

Atlas based segmentation