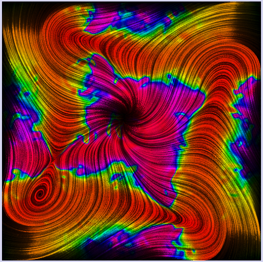
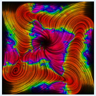
Boundary Conditions for Curvature Based Registration



Jens-Peer Kuska

18. Mai 2005

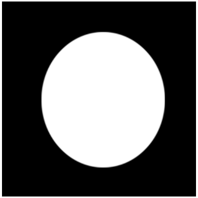
Boundary Conditions for Curvature Based Registration



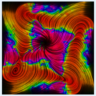
Jens-Peer Kuska

What does the registration ?

The goal of image registration is to find a transformation for the image I so that it matches the image R.



$$\vec{u}(x,y) \mapsto$$



Jens-Peer Kuska

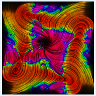
Kinds of Registration

Parametric Registration

- rigid registration, find translation, rotation and scaling to match the images
- polynomial registration find a nonlinear transformation in form of a polynomial

Nonparametric Registration

- elasic registration, the image transformed like a elasic membrane
- fluid registration, a fluid like deformation
- curvature based registration



Jens-Peer Kuska

Curvature Based Registration

For the template image $I(\vec{x})$ and the registration image $R(\vec{x})$ one has to find a displacement field $\vec{u}(\vec{x})$ so that

$$\mathcal{J}[\vec{u}] = \mathcal{D}[\vec{u}] + \alpha\, \mathcal{S}^{\mathrm{curv}}[\vec{u}] \tag{1}$$

has a minimum

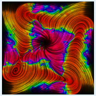
$$\min_{\vec{u}} \mathcal{J}[\vec{u}], \qquad \vec{u} : \mathbb{R}^2 \to \mathbb{R}^2 \tag{2}$$

with

$$\mathcal{D}[\vec{u}] = \frac{1}{2} \int_\Omega \left(I(\vec{x} - \vec{u}(\vec{x})) - R(\vec{x}) \right)^2 \, d\vec{x} \tag{3}$$

$$\mathcal{S}^{\mathrm{curv}}[\vec{u}] = \frac{1}{2} \int_\Omega (\Delta \vec{u})^T \cdot (\Delta \vec{u}) \, d\vec{x} \tag{4}$$

with the two dimensional Laplace operator Δ.



Jens-Peer Kuska

Equation for the displacement field

$$\alpha\,\Delta^2\vec{u} - \vec{f}(\vec{x},\vec{u}) = 0 \tag{5}$$

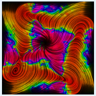
with

$$\vec{f}(\vec{x},\vec{u}) = \left[R(\vec{x}) - I(\vec{x}-\vec{u}(\vec{x}))\right]\nabla I(\vec{x}-\vec{u}(\vec{x})) \tag{6}$$

This equation is very hard to solve direct and one introduces an artificial time t:

$$\frac{\partial\vec{u}}{\partial t}(\vec{x},t) + \alpha\,\Delta^2\vec{u}(\vec{x},t) = \vec{f}(\vec{x},\vec{u}(\vec{x},t)) \tag{7}$$

this equation converges for $t \to \infty$ to the solution of the static equation.



Jens-Peer Kuska

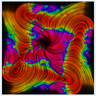
Boundary Conditions

We consider three types of boundary conditions for this fourth order equation in the space variable $\vec{x}$:

a) periodic boundary conditions

b) vanishing first and third derivative in normal direction to the boundary

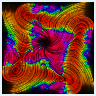
c) no displacement on the boundary and vanishing second derivative on the boundary.



Jens-Peer Kuska

Global Transformations and the Boundary Conditions

The cases a) and b) have the obvious advantage, that a rigid translation of the I, i.e., $\vec{u}(\vec{x}) = \vec{m}$ with a constant vector $\vec{m}$, does not violate the boundary conditions and fulfill the homogenous equation (5).
None of the boundary condtions is consistent with a rigid transformation that include a rotation around $\vec{c}$ of the image, i.e, $\vec{u}(\vec{x}) = \overleftrightarrow{R}(\vec{x} - \vec{c}) + \vec{m}$ with the rotation matrix $\overleftrightarrow{R}$.



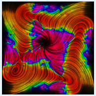
Jens-Peer Kuska

Multiresolution Formulation

Since the solution is essential a variational problem, one has to avoid, that the time depend equation converges to a local minimum of the variational equation for $\vec{u}$.
Since we operate on digital images we have to determine the discret solution $\vec{U}$ in a regular mesh with

$$\vec{U}^{(l)}(t) = \vec{u}(\vec{x}_{i,j}, t) \qquad i = 1, \ldots N^{(l)}, j = 1, \ldots, M^{(l)}$$

The superscript l denote a certain discretisation level, were $l = 0$ is the coarsest grid an $l = L$ the finest gird.



Jens-Peer Kuska

Switch Between Different Resolution Levels

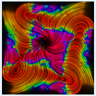
We will introduce the operator

$$\mathcal{R}\left[\vec{U}^{(l+1)}(t)\right] = \vec{U}^{(l)}(t)$$

for the restriction of solution $\vec{U}$ to the coarse grid and the operator

$$\mathcal{P}\left[\vec{U}^{(l)}(t)\right] = \vec{U}^{(l+1)}(t)$$

for the prolongation to the finer level.



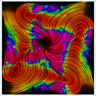
Jens-Peer Kuska

Iteration in Time: From Coarse Grid $\mapsto$ Fine

The time depend equation (7) on the level $l = 0$ is discretized with the time step h by an implicit mid-point rule for the linear operator Δ^2 and by a explicit Euler rule for the nonlinear term $\vec{f}$

$$\vec{V}^{(0)}(t+h) - \vec{U}^{(0)}(t) = \\ -\frac{\alpha\, h}{2}\left(\Delta^2 \vec{V}^{(0)}(t+h) + \Delta^2 \vec{U}^{(0)}(t)\right) + h\, \vec{F}^{(0)}(\vec{U}^{(0)}(t))$$

where $\vec{F}^{(0)}$ is the discrete approximation of $\vec{f}$ on the mesh points of level $l = 0$. And $\vec{V}^{(l)}$ is a temporary approximation for $\vec{U}^{(l)}$.

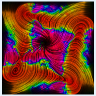


Jens-Peer Kuska

Than the finer levels $l > 0$ computed by the the equation

$$\begin{aligned}
&\vec{V}^{(l)}(t+h) - \vec{U}^{(l)}(t) = \\
&\quad -\frac{\alpha\, h}{2}\left(\Delta^2 \vec{V}^{(l)}(t+h) + \Delta^2 \vec{U}^{(l)}(t)\right) \\
&\qquad + h\, \vec{F}^{(l)}\left(\frac{1}{2}(\mathcal{P}[\vec{V}^{(l-1)}](t+1) + \vec{U}^{(l)}(t))\right)
\end{aligned}$$

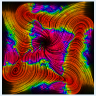
This equation uses the implicit mid-point rule also in the nonlinear term of $\vec{F}$, but instead of an iteration of $\vec{V}^{(l)}$ the prolongation of $\vec{V}^{(l-1)}$ from the finer level is used as approximation to $\vec{V}^{(l)}$.



Iteration in Time: From Fine Grid $\mapsto$ Coarse

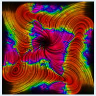
hen $\vec{V}^{(L)}(t+h)$ on the finest level is computed the new $\vec{U}^{(l)}(t+h)$ are determined by

$$\begin{aligned} \vec{U}^{(L)} &= \vec{V}^{(L)} \\ \vec{U}^{(l)} &= \mathcal{R}\left[\vec{U}^{(l+1)}\right] \qquad l = L-1, \ldots, 0. \end{aligned}$$



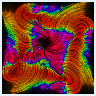
Remarks

- The time integration is stable for all time steps h.
- The multiresolution formulation is used to handle the nonlinearity in $\vec{f}$ and not to speed up the solution for a single resolution level



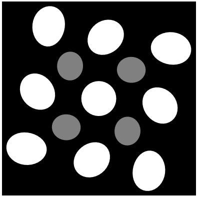
Discrete Solution on a Single Level

The differential operator is approximated by a finite difference expression and the resulting finite difference equation is solved by a discrete Fourier transform. For periodic boundary conditions a normal DFT ist used, for zero displacement boundary conditions a DST (DST-2) is used and for zero derivative boundary conditions a DCT (DCT-2) will be used.

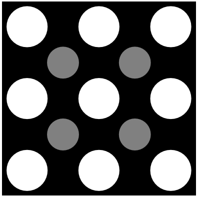


Jens-Peer Kuska

Example



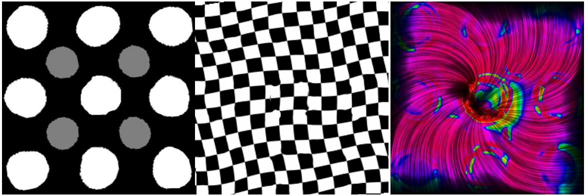
$$\vec{u}(x,y)$$
$$\mapsto$$



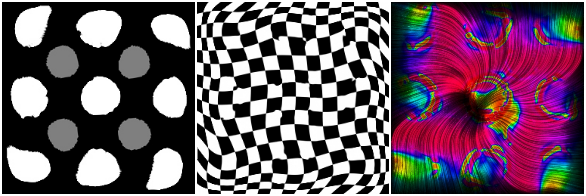
- global rotation
- local shape distortions
- several local minima for the registration functional

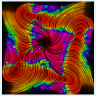
Registration Results

$\partial_{\vec{n}}\vec{u}|_{\vec{x}\in\Gamma} = 0$



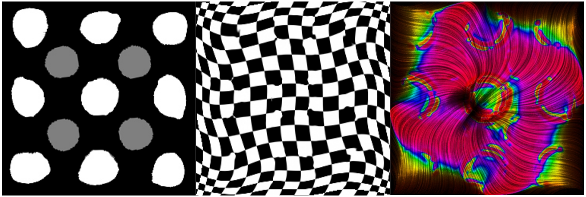
$\vec{u}|_{\vec{x}\in\Gamma} = 0$



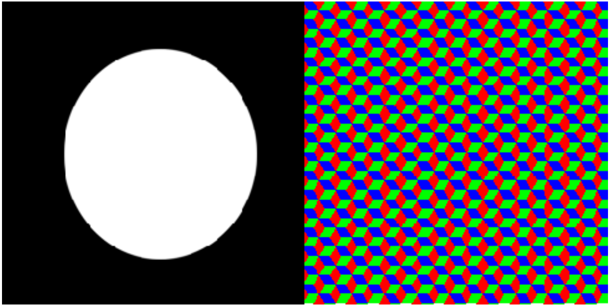


Jens-Peer Kuska

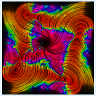
periodic:



Zero Boundary

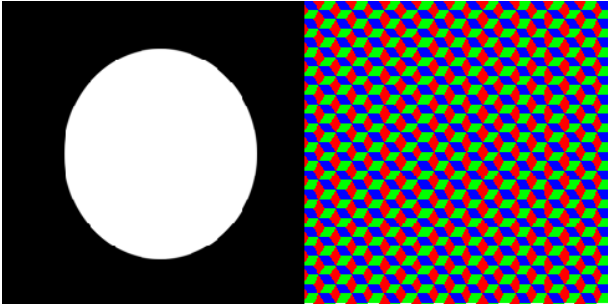


Transition form circle to C with zero boundary conditions

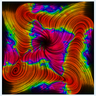


Jens-Peer Kuska

Zero Derivative

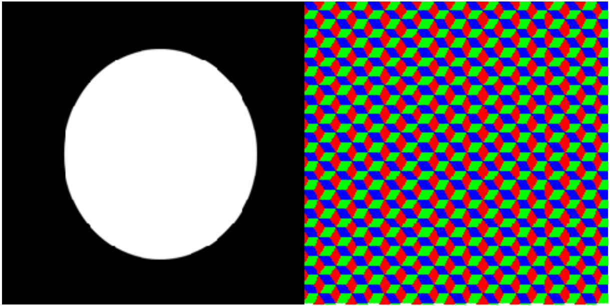


Transition form circle to C with zero derivative boundary conditions

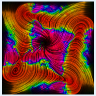


Jens-Peer Kuska

Periodic



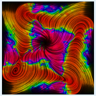
Transition form circle to C with periodic boundary conditions



Jens-Peer Kuska

Final Remarks

- the proper choice of the boundary conditions depends on the images
- the boundary conditions influence the convergence speed
- since real Fourier transforms work with real data the zero and zero derivative boundary conditions are two times faster



Jens-Peer Kuska