

Modellierung für Bildverarbeitung

Stephen Keeling

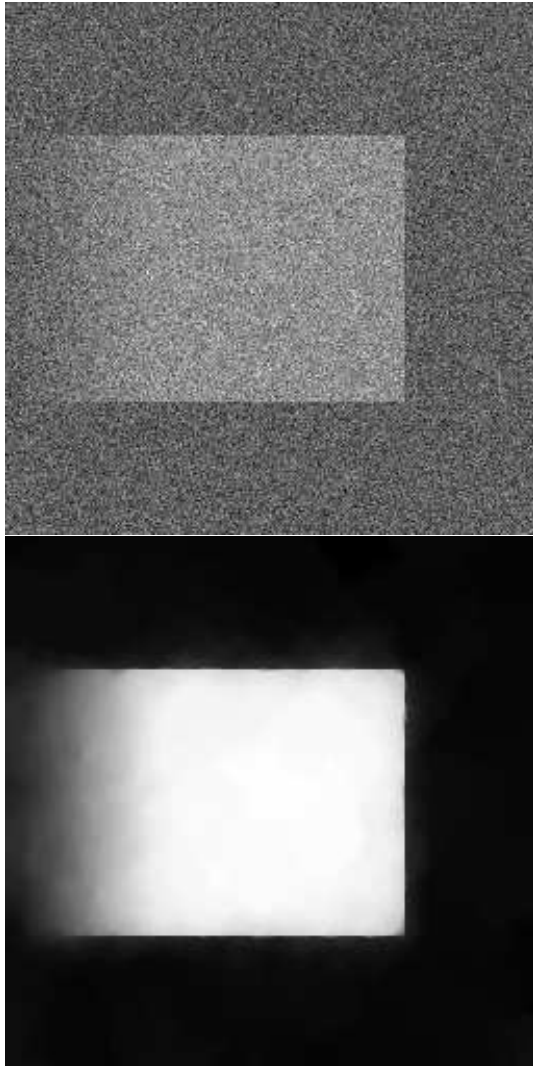
Institut für Mathematik und Wissenschaftliches Rechnen

Überblick

- Einführung in die Bildverarbeitung.
- Modellierung von natürlichen und künstlichen Systemen.
- **Entrauschen**: Erklärung und Erprobung mit **MATLAB** Code.
- **Segmentierung**: Erklärung und Erprobung mit **MATLAB** Code.
- **Registrierung**: Erklärung und Erprobung mit **MATLAB** Code.
- Werkzeuge für dynamische Systeme: **NETLOGO** und **VENSIM**.

Einführung in die Bildverarbeitung

Entrauschen



Segmentierung

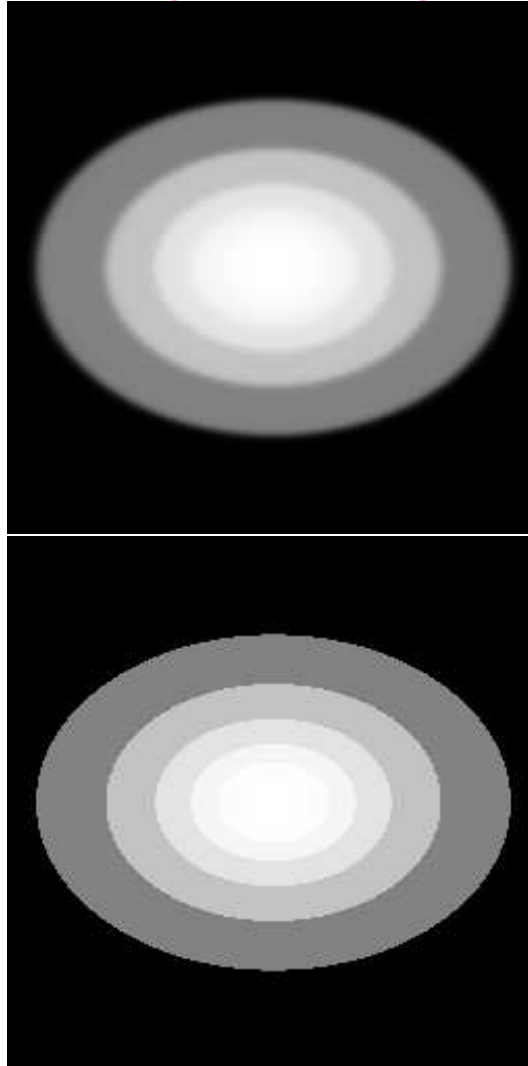
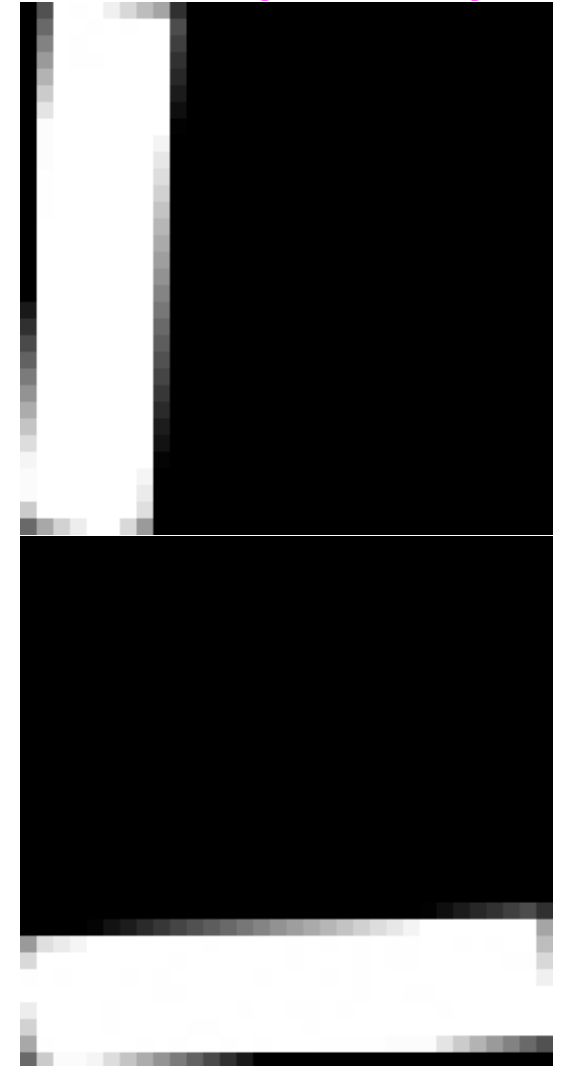


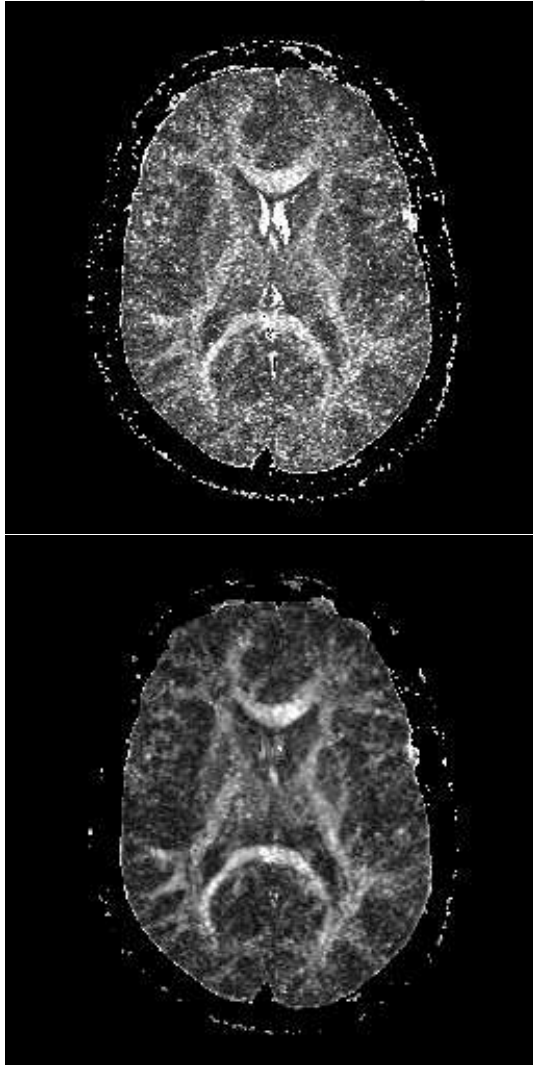
Bild-Registrierung



Anwendungen in der MRI

Entrauschen:

Diffusion Tensor Imaging,
Fiber-Tracking



Segmentierung:

Volumenrechnung,
Inhomogenitätskorrektur

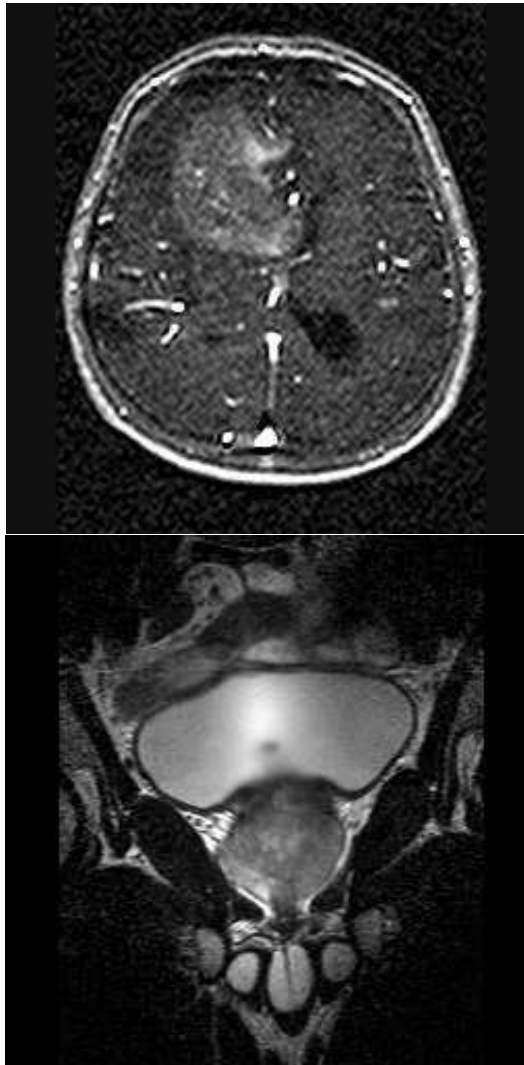
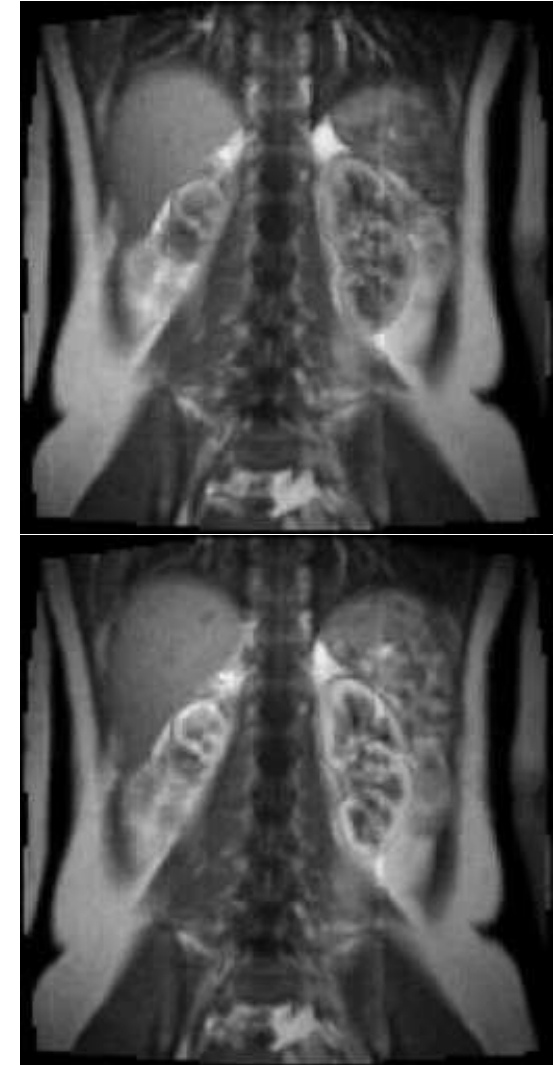


Bild-Registrierung:

Physiologische Bewegung
mit Kontrastmittel



Natürliche und Künstliche Systeme

- Beispiel 1: Auslenkung einer beladenen Membran in Gleichgewicht.
- Beispiel 2: Entrauschen eines Bilds.

- Mathematische Formulierungen sind fast identisch.
- Bildverarbeitungsprobleme sind von der Natur nicht eingeschränkt!

Beispiel 1: Auslenkung einer beladenen Membran oder Saite (1D) in Gleichgewicht

f = äußere senkrecht wirkende (beladende) Kraft

u = Auslenkung der waagerechten Saite

$x \in [-1, +1]$ = Intervall der Saite

T = Spannung der Saite

$\int_{-1}^{+1} [\sqrt{1 + u'^2} - 1] dx$ = Längeänderung der ausgelenkten Saite

E = potentielle Energie (P.E.) der Saite

= Spannung \times Längeänderung $-$ Kraft \times Auslenkung

$$E(u) = \underbrace{T \int_{-1}^{+1} [\sqrt{1 + u'^2} - 1] dx}_{\substack{\text{Zunahme der P.E.} \\ \text{wegen innerer} \\ \text{elastischer Kräfte}}} - \underbrace{\int_{-1}^{+1} f u dx}_{\substack{\text{Abnahme der P.E.} \\ \text{wegen äußerer} \\ \text{angewandter Kräfte}}}$$

$$|u'| \text{ klein} \quad \Rightarrow \quad E(u) = \frac{1}{2} T \int_{-1}^{+1} u'^2 dx - \int_{-1}^{+1} f u dx$$

Minimierung der Potentiellen Energie

$$E(u + \varepsilon v) = \frac{1}{2}T \int_{-1}^{+1} (u' + \varepsilon v')^2 dx - \int_{-1}^{+1} (u + \varepsilon v) f dx$$

$$D_\varepsilon E(u + \varepsilon v) = T \int_{-1}^{+1} (u' + \varepsilon v') v' dx - \int_{-1}^{+1} v f dx$$

$$\lim_{\varepsilon \rightarrow 0} D_\varepsilon E(u + \varepsilon v) = T \int_{-1}^{+1} u' v' dx - \int_{-1}^{+1} v f dx$$

$$\frac{\delta E}{\delta u}(u; v) = T \left[- \int_{-1}^{+1} u'' v dx + u' v \Big|_{-1}^{+1} \right] - \int_{-1}^{+1} v f dx$$

$$= \int_{-1}^{+1} [-T u'' - f] v dx + T u' v \Big|_{-1}^{+1}$$

$$\frac{\delta E}{\delta u}(u; v) = 0, \quad \forall v \quad \Leftrightarrow \quad \begin{cases} -T u'' = f, & x \in (-1, +1) \\ u' = 0, & x = -1, +1 \end{cases}$$

Beispiel 2: Entauschen eines Bilds

\tilde{u} = verauschtes Bild

u = entauschte Abschätzung

$x \in [-1, +1]$ = Intervall der Bilder

μ = künstliche Spannung

E = künstliche potentielle Energie (P.E.)

$$E(u) = \underbrace{\int_{-1}^{+1} [u - \tilde{u}]^2 dx}_{\text{Abnahme der P.E. durch Anziehung gegen } \tilde{u}} + \underbrace{\mu \int_{-1}^{+1} u'^2 dx}_{\text{Zunahme der P.E. durch Schwingungen in } u}$$

$$\frac{\delta E}{\delta u}(u; v) = 2 \int_{-1}^{+1} [u - \tilde{u} - \mu u''] v dx + \mu u' v \Big|_{-1}^{+1}$$

$$\frac{\delta E}{\delta u}(u; v) = 0, \quad \forall v \quad \Leftrightarrow \quad \begin{cases} -\mu u'' + u = \tilde{u}, & x \in (-1, +1) \\ u' = 0, & x = -1, +1 \end{cases}$$

Vergleich der Beispiele

- Beispiel 1: Auslenkung einer beladenen Membran in Gleichgewicht.

$$\begin{cases} -Tu'' = f, & x \in (-1, +1) \\ u' = 0, & x = -1, +1 \end{cases}$$

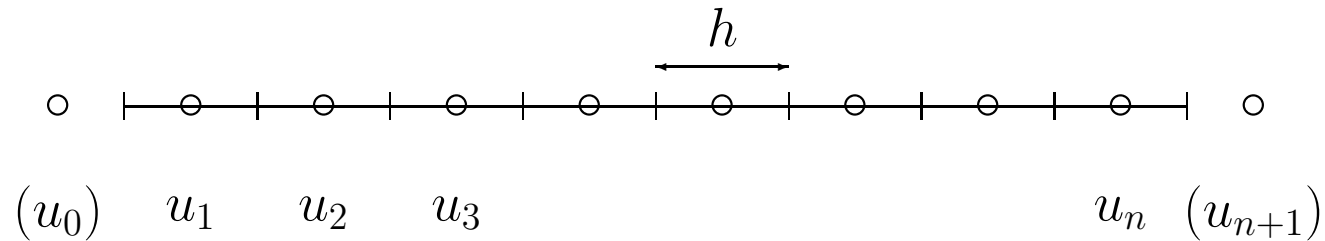
- Beispiel 2: Entauschen eines Bilds.

$$\begin{cases} -\mu u'' + u = \tilde{u}, & x \in (-1, +1) \\ u' = 0, & x = -1, +1 \end{cases}$$

Entrauschen: Erklärung und Erprobung mit MATLAB Code

$$\begin{cases} -\mu u'' + u = \tilde{u}, & x \in (-1, +1) \\ u' = 0, & x = -1, +1 \end{cases}$$

Gitter von Zellen:



Diskretisierung:

$$-u''_2 \approx -\frac{\frac{u_3 - u_2}{h} - \frac{u_2 - u_1}{h}}{h} = \frac{-u_3 + 2u_2 - u_1}{h^2}$$

$$-u''_1 \approx -\frac{\frac{u_2 - u_1}{h} - \frac{u_1 - u_0}{h}}{h} \Big|_{u'=0} = \frac{-u_2 + u_1}{h^2}$$

$$-u''_n \approx -\frac{\frac{u_{n+1} - u_n}{h} - \frac{u_n - u_{n-1}}{h}}{h} \Big|_{u'=0} = \frac{u_n - u_{n-1}}{h^2}$$

Entauschen: Erklärung und Erprobung mit MATLAB Code

$$\begin{cases} -\mu u'' + u = \tilde{u}, & x \in (-1, +1) \\ u' = 0, & x = -1, +1 \end{cases}$$

$$-u'' \longrightarrow \frac{1}{h^2} \begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \cdots & \cdots & \cdots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} = \frac{1}{h^2} D \mathbf{u}, \quad \text{Daten: } \begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \vdots \\ \vdots \\ \tilde{u}_{n-1} \\ \tilde{u}_n \end{bmatrix} = \tilde{\mathbf{u}}$$

$$-\mu u'' + u \longrightarrow \left[\frac{\mu}{h^2} D + I \right] \mathbf{u} = A \mathbf{u}$$

$$\text{Löse: } A \mathbf{u} = \tilde{\mathbf{u}}$$

Siehe Code: `entrausch.m`

entrausch.m

```
function [u,x] = entrausch(n,pn,mu)

% das Gitter
h = 2/n; % Zellenbreite
x = -1.0 + 0.5*h + h*(0:(n-1))'; % Zellzentren

% verrauschte Funktion
k = fix(n/2);
v0 = [zeros(k,1);ones(n-k,1)]; % Treppenfunktion

% v0 = sin(10*x); % mehrere Treppen
% v0 = double(v0 > 0) - double(v0 <= 0);

randn('seed',0); % init randn
v = v0 + pn*randn(n,1); % verrauschte Funktion

% Diskretisierung des entrauschenden Operators
D = diag([ones(n-1,1);0])... % Diskretisierung
+ diag([0;ones(n-1,1)])... % der zweiten
- diag(ones(n-1,1),1)... % Ableitung mit
- diag(ones(n-1,1),-1); % Randbedingungen
A = eye(n) + mu*D/h^2; % entrauschender Operator

% loese das System
u = A\v; % entrauschte Funktion

% zeige exakt, verrauscht und entrauscht zusammen
plot(x,v0,'b',x,v,'r',x,u,'g'); % grafisch darstellen
```

Verbesserte Potentielle Energie

Suche Funktion besser als $\phi(s) = s$:

$$E(u) = \int_{-1}^{+1} [u - \tilde{u}]^2 dx + \int_{-1}^{+1} \phi(u'^2) dx$$

$$\frac{\delta E}{\delta u}(u; v) = 2 \int_{-1}^{+1} \{u - \tilde{u} - [\phi'(u'^2)u']'\} v dx + 2 [\phi'(u'^2)u'v]_{-1}^{+1}$$

$$\frac{\delta E}{\delta u}(u; v) = 0, \quad \forall v \quad \Leftrightarrow \quad \begin{cases} -[\phi'(u'^2)u']' + u = \tilde{u}, & x \in (-1, +1) \\ u' = 0, & x = -1, +1 \end{cases}$$

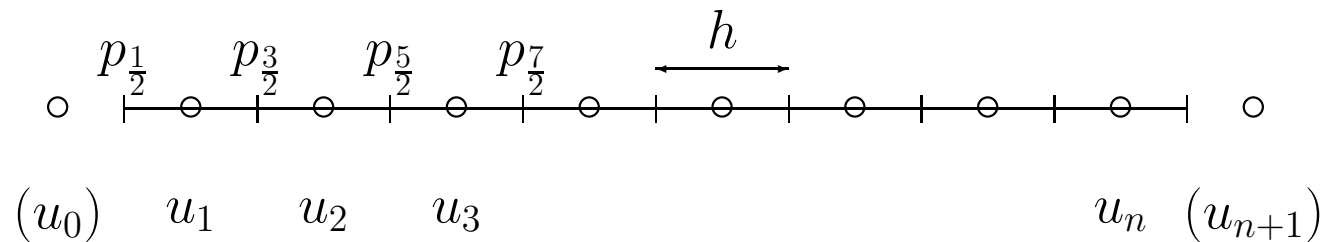
Picard Iteration zur Lösung:

$$\begin{cases} -[\phi'(u_k'^2)u'_{k+1}]' + u_{k+1} = \tilde{u}, & x \in (-1, +1) \\ u'_{k+1} = 0, & x = -1, +1 \end{cases}$$

Erprobung mit Verbessertem MATLAB Code

$$\begin{cases} -[pu']' + u = \tilde{u}, & x \in (-1, +1) \\ u' = 0, & x = -1, +1 \end{cases}$$

Variable Leitwerte in Schnittstellen:



Diskretisierung:

$$-[pu'_2]' \approx -\frac{p_{\frac{5}{2}} \frac{u_3 - u_2}{h} - p_{\frac{3}{2}} \frac{u_2 - u_1}{h}}{h}$$

$$-[pu'_1]' \approx -\frac{p_{\frac{3}{2}} \frac{u_2 - u_1}{h} - p_{\frac{1}{2}} \frac{u_1 - u_0}{h}}{h} \Big|_{u'=0}$$

$$-[pu'_n]' \approx -\frac{p_{\frac{2n+1}{2}} \frac{u_{n+1} - u_n}{h} - p_{\frac{2n-1}{2}} \frac{u_n - u_{n-1}}{h}}{h} \Big|_{u'=0}$$

entrausch_nl.m

```
function [u,xc] = entrausch_nl(n,pn,mu)

% das Gitter
h = 2/n; % Zellenbreite
xc = -1.0 + 0.5*h + h*(0:(n-1))'; % Zellzentren
xg = -1.0 + h*(0:n)'; % Zellschnittstellen

% verrauschte Funktion
k = fix(n/2);
v0 = [zeros(k,1);ones(n-k,1)]; % Treppenfunktion

% v0 = sin(10*xc); % mehrere Treppen
% v0 = double(v0 > 0) - double(v0 <= 0);

randn('seed',0); % init randn
v = v0 + pn*randn(n,1); % verrauschte Funktion

% Begin mit linearer Abschaetzung
D = diag([ones(n-1,1);0])... % Diskretisierung
+ diag([0;ones(n-1,1)])... % der zweiten
- diag(ones(n-1,1),1)... % Ableitung mit
- diag(ones(n-1,1),-1); % Randbedingungen
A = eye(n) + mu*D/h^2; % entrauschender Operator
u = A\v; % entrauschte Funktion
plot(xc,v0,'b',xc,v,'r',xc,u,'g'); % grafisch darstellen

% Picard Iteration: setz fort bis Unterschiede klein genug
itmax = 100; % max Iterationen
tol = 1.0e-4; % Abbruchkriterium
omega = 1.0; % Daempfung
it = 0;
us = v;
while ((max(abs(u-us)) > tol*max(abs(u))) & (it < itmax))
    us = u; % speichere altes u
    ux = (u(2:n)-u(1:(n-1)))/h; % ux in Zellschnittstellen
    ps = phistrich(ux.^2); % Leitwerte
    D = diag([ps;0])... % Diskretisierung des
+ diag([0;ps])... % nicht linearen
- diag(ps,+1)... % entrauschenden
- diag(ps,-1); % Operators mit
    A = eye(n) + mu*D/h^2; % Randbedingungen
    u = A\v; % aktuelle Abschaetzung
    u = us + omega*(u - us); % Daempfung
    it = it+1;
    plot(xc,v0,'b',xc,v,'r',xc,u,'g'); % grafisch darstellen
    drawnow;
end

% zeige exakt, verrauscht und entrauscht zusammen
plot(xc,v0,'b',xc,v,'r',xc,u,'g'); % grafisch darstellen

% Leitwertsfunktion: Energieterm ist phi(ux^2).
function ps = phistrich(ux2)
q = 1; % phi(s)=[s+ep]^(q/2)
ep = 1.0e-2; % phi'(s)=
ps = (q/2)./(ux2 + ep).^(-1-q/2); % (q/2)[s+ep]^(q/2-1)
```

Segmentierung: Erklärung und Erprobung mit MATLAB Code

\tilde{u} = verrauschtes Bild

u = entrauschte Abschätzung

$x \in \Omega = [-1, +1]$ = Intervall der Bilder

$\Gamma \subset \Omega$ = Kanten

μ = künstliche Spannung in $\Omega \setminus \Gamma$

E = künstliche potentielle Energie (P.E.)

$$E(u) = \underbrace{\int_{\Omega} [u - \tilde{u}]^2 dx}_{\substack{\text{Abnahme der P.E.} \\ \text{durch Anziehung gegen } \tilde{u}}} + \underbrace{\mu \int_{\Omega \setminus \Gamma} u'^2 dx}_{\substack{\text{Zunahme der P.E.} \\ \text{durch Schwingungen in } u}}$$

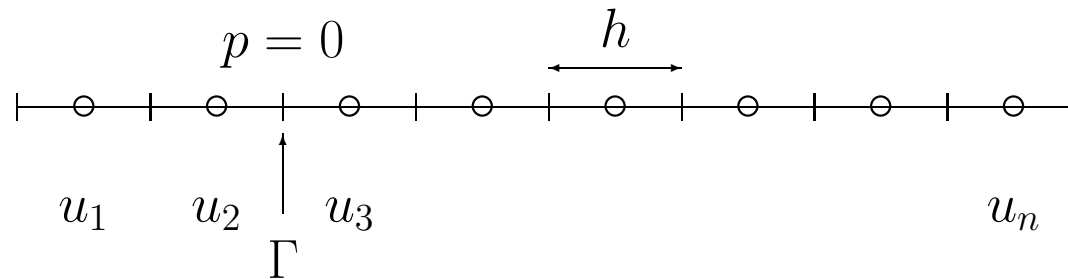
$$\frac{\delta E}{\delta u}(u; v) = 2 \int_{\Omega \setminus \Gamma} [u - \tilde{u} - \mu u''] v dx + \mu u' v|_{\Gamma, \partial \Omega}$$

$$\frac{\delta E}{\delta u}(u; v) = 0, \quad \forall v \quad \Leftrightarrow \quad \begin{cases} -\mu u'' + u = \tilde{u}, & x \in \Omega \setminus \Gamma \\ u' = 0, & x \in \Gamma, \partial \Omega \end{cases}$$

Segmentierung: Erklärung und Erprobung mit MATLAB Code

$$\begin{cases} -\mu u'' + u = \tilde{u}, & x \in \Omega \setminus \Gamma \\ u' = 0, & x \in \Gamma, \partial\Omega \end{cases}$$

Leitwert Null in Kante:



Diskretisierung:

$$\begin{bmatrix} 1 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & -1 & 2 & -1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 1 \end{bmatrix} \longrightarrow D = \begin{bmatrix} 1 & -1 & & & & & \\ -1 & 1 & & & & & \\ & & & 1 & -1 & & \\ & & & -1 & 2 & -1 & \\ & & & & \ddots & \ddots & \ddots \\ & & & & & -1 & 2 & -1 \\ & & & & & & -1 & 1 \end{bmatrix}$$

$$-\mu u'' + u \longrightarrow \left[\frac{\mu}{h^2} D + I \right] \mathbf{u} = \mathbf{A} \mathbf{u}$$

Löse: $\mathbf{A} \mathbf{u} = \tilde{\mathbf{u}}$, Siehe Code: `segment.m`

segment.m

```
function [u,x] = segment(n,pn,mu)

% das Gitter
h = 2/n; % Zellenbreite
x = -1.0 + 0.5*h + h*(0:(n-1))'; % Zellzentren

% verrauschte Funktion
k = fix(n/2);
v0 = [zeros(k,1);ones(n-k,1)]; % Treppenfunktion

% v0 = sin(10*x); % mehrere Treppen
% v0 = double(v0 > 0) - double(v0 <= 0);

randn('seed',0); % init randn
v = v0 + pn*randn(n,1); % verrauschte Funktion

% Diskretisierung des entauschenden Operators, Bruch in i-ter Komponente
i = v0(2:n)-v0(1:(n-1)); % Ableitung exakter Loesung
i = find(i ~= 0); % nicht null, dann ist
ps = ones(n-1,1); % Leitwert null:
ps(i) = 0;
D = diag([ps;0])... % Diskretisierung
+ diag([0;ps])... % der zweiten
- diag(ps,+1)... % Ableitung mit
- diag(ps,-1); % Randbedingungen
A = eye(n) + mu*D/h^2; % entauschender Operator

% loese das System
u = A\v; % entauschte Funktion

% zeige exakt, verrauscht und entauscht zusammen
plot(x,v0,'b',x,v,'r',x,u,'g'); % grafisch darstellen
```

Segmentierung mit Kanten durch *Level Sets* definiert

- \tilde{u} = verrauschtes Bild
- $x \in \Omega = [-1, +1]$ = Intervall der Bilder
- $\Gamma \subset \Omega$ = Kanten
- ϕ = *Level Set* Funktion, $\phi|_{\Gamma} = 0$, $\phi|_{\Gamma_+} > 0$, $\phi|_{\Gamma_-} < 0$
- c_1, c_2 = durchschnittliche Werte von \tilde{u} in $\{\phi < 0\}$ bzw. $\{\phi > 0\}$
- H = *Heaviside* Treppenfunktion, $H(s) = [\text{Signum}(s) + 1]/2$
- u = $c_1 H(-\phi) + c_2 H(\phi)$, segmentierte Abschätzung
- μ = künstliche Spannung in $\Omega \setminus \Gamma$
- E = künstliche potentielle Energie (P.E.)

$$E(\phi, c_1, c_2) = \underbrace{\int_{\Omega} |H(\phi)'| dx}_{\text{Zunahme der P.E. durch Anzahl der Kanten}} + \underbrace{\mu \int_{\Omega} [H(-\phi)(c_1 - \tilde{u})^2 + H(\phi)(c_2 - \tilde{u})^2] dx}_{\text{Abnahme der P.E. durch Anziehung gegen } \tilde{u}}$$

$$H(\phi) = 0 \text{ in } \{\phi < 0\}, \quad H(-\phi) = 0 \text{ in } \{\phi > 0\}, \quad H(\phi)' \neq 0 \text{ in } \{\phi = 0\}$$

Segmentierung mit Kanten definiert durch *Level Sets*

$$\frac{\partial E}{\partial c_1} = 0 \Leftrightarrow c_1 = \int_{\phi < 0} \tilde{u} dx / \int_{\phi < 0} dx, \quad \frac{\partial E}{\partial c_2} = 0 \Leftrightarrow c_2 = \int_{\phi > 0} \tilde{u} dx / \int_{\phi > 0} dx$$

$$\frac{\delta E}{\delta \phi}(\phi; \psi) = \int_{\Omega} H'(\phi) \left\{ - \left(\frac{\phi'}{|\phi'|} \right)' - \mu(c_1 - \tilde{u})^2 + \mu(c_2 - \tilde{u})^2 \right\} \psi dx$$

$$\frac{\delta E}{\delta \phi}(\phi; \psi) = 0, \forall \psi \Leftrightarrow H'(\phi) \left[- \left(\frac{\phi'}{|\phi'|} \right)' - \mu(c_1 - \tilde{u})^2 + \mu(c_2 - \tilde{u})^2 \right] = 0, x \in \Omega$$

Evolution gegen Gleichgewicht:

$$\phi_{k+1} - \phi_k = \tau \left[\left(\frac{\phi'_k}{|\phi'_k|} \right)' + \mu(c_1 - \tilde{u})^2 - \mu(c_2 - \tilde{u})^2 \right]$$

$$c_1 = \int_{\phi_k < 0} \tilde{u} dx / \int_{\phi_k < 0} dx$$

$$c_2 = \int_{\phi_k > 0} \tilde{u} dx / \int_{\phi_k > 0} dx$$

Siehe Code: `segment_ls.m`

segment_ls.m

```
function [u,x] = segment_ls(n,pn,mu)

% das Gitter
h = 2/n; % Zellenbreite
x = -1.0 + 0.5*h + h*(0:(n-1))'; % Zellzentren

% verrauschte Funktion
k = fix(n/2);
v0 = [zeros(k,1);ones(n-k,1)]; % Treppenfunktion

% v0 = sin(10*x); % mehrere Treppen
% v0 = double(v0 > 0) - double(v0 <= 0);

randn('seed',0); % init randn
v = v0 + pn*randn(n,1); % verrauschte Funktion

% Begin mit einfacher Level-Set Funktion
p = -1 + 2*abs(x);

% Evolution der Level-Sets
itmax = 100; % max Iterationen
dt = 1.0e1; % Zeitschrittbreite
for it=1:itmax
    In = find(p < 0); % innere Menge
    Au = find(p >= 0); % aeussere Menge

    c1 = sum(v(In))/length(v(In)); % innerer Wert
    c2 = sum(v(Au))/length(v(Au)); % aeusserer Wert

    dp = p;
    dp = [2*dp(1)-dp(2);dp;2*dp(n)-dp(n-1)];
    dp = (dp(3:(n+2))-dp(1:n))/(2*h);
    dp = dp./(dp + (dp == 0));
    dp = [2*dp(1)-dp(2);dp;2*dp(n)-dp(n-1)];
    dp = (dp(3:(n+2))-dp(1:n))/(2*h); % (p'/|p'|)'

    p = p + dt*(dp + mu*(c1-v).^2 - mu*(c2-v).^2); % aktuelle LS Funktion

    u1 = zeros(n,1);
    u1(In) = 1; % u1=1 in innerer Menge
    u2 = zeros(n,1);
    u2(Au) = 1; % u2=1 in aeusserer Menge
    u = c1*u1 + c2*u2; % segmentierte Funktion
    plot(x,v0,'b',x,v,'r',x,u,'g'); % grafisch darstellen
    drawnow;

    it = it+1;
end

% zeige exakt, verrauscht und entrauscht zusammen
plot(x,v0,'b',x,v,'r',x,u,'g'); % grafisch darstellen
```

Registrierung: Erklärung und Erprobung mit MATLAB Code

I_0 = (verraushtes) Bild 0

I_1 = (verraushtes) Bild 1

$x \in [-1, +1]$ = Intervall der Bilder

u = Auslenkung sodass $I_0(x) \approx I_1(u(x))$

$I(x, t) = I_1(x + t(u(x) - x))$ interpoliertes Bild

$I(x, 0) = I_1(x), I(x, 1) = I_1(u(x)) \approx I_0(x)$

μ = künstlicher Biegungsparameter

E = künstliche potentielle Energie (P.E.)

$$E(u) = \underbrace{\int_{-1}^{+1} |I_0(x) - I_1(u(x))|^2 dx}_{\text{Abnahme der P.E. durch Anziehung } I_1(u) \text{ gegen } I_0} + \underbrace{\mu \int_{-1}^{-1} u''^2 dx}_{\text{Zunahme der P.E. durch Biegung in } u}$$

$$\frac{\delta E}{\delta u}(u; v) = 2 \int_{-1}^{+1} \{ [I_1(u) - I_0] I_1'(u) + \mu u'''' \} v dx - \mu u'' v' |_{-1}^{+1} + \mu u''' v |_{-1}^{+1}$$

$$\frac{\delta E}{\delta u}(u; v) = 0 \Leftrightarrow \begin{cases} \mu u'''' + [I_1(u) - I_0] I_1'(u) = 0, & x \in (-1, 1) \\ u'' = u''' = 0, & x = -1, +1 \end{cases}$$

regis.m

```
function [u,x] = regis(n,pn,mu)

% das Gitter
h = 2/n; % Zellenbreite
x = -1.0 + 0.5*h + h*(0:(n-1))'; % Zellzentren

% Funktionen zu registrieren
k1 = fix(n/8);
k2 = fix(3*n/8);
I0 = sin((1:(n-k1-k2))/10)';
I0 = [zeros(k1,1);I0;zeros(k2,1)]; % 1. Funktion
k1 = fix(3*n/8);
k2 = fix(n/8);
I1 = sin((1:(n-k1-k2))/10)';
I1 = [zeros(k1,1);I1;zeros(k2,1)]; % 2. Funktion

% verrauschte Funktionen
randn('seed',0); % init randn
I0 = I0 + pn*randn(n,1); % 1. verrauschte Funktion
I1 = I1 + pn*randn(n,1); % 2. verrauschte Funktion

% Begin mit Identitaet
u = x;

% Diskretisierung der vierten Ableitung mit Randbedingungen
D = 6*diag(ones(n ,1) ) ...
- 4*diag(ones(n-1,1),-1) ...
- 4*diag(ones(n-1,1),+1) ...
+ diag(ones(n-2,1),-2) ...
+ diag(ones(n-2,1),+2);

D( 1, 1) = 1;
D( n, n) = 1;

D( 1, 2) = -2;
D( 2, 1) = -2;
D( n,n-1) = -2;
D(n-1, n) = -2;

D( 2, 2) = 5;
D(n-1,n-1) = 5;

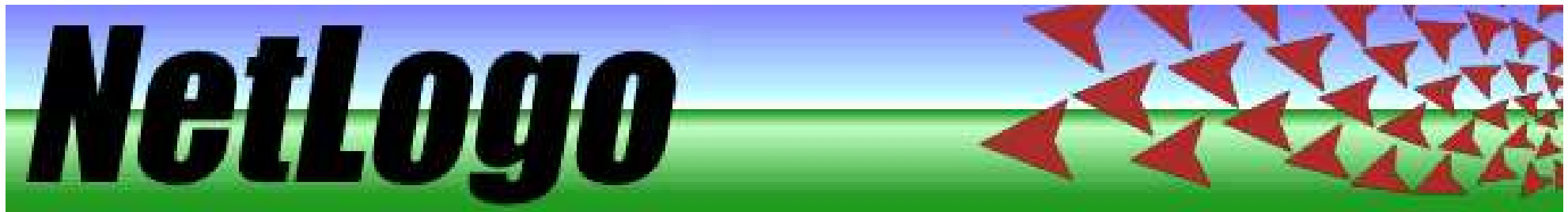
% Ableitung von I1 in x
Ix = [2*I1(1)-I1(2);I1;2*I1(n)-I1(n-1)];
Ix = (Ix(3:(n+2))-Ix(1:n))/h;

% Quasi-Newton Iteration: setze fort bis Unterschiede klein genug
itmax = 1000; % max Iterationen
tol = 1.0e-4; % Abbruchkriterium
omega = 1.0; % Daempfung
it = 0;
us = u+1;
while ((max(abs(u-us)) > tol*max(abs(u))) & (it < itmax))
    us = u; % speichere altes u
    Is = interp1(x,Ix,u,'linear','extrap'); % Ableitung von I1 in u
    Iu = interp1(x,I1,u,'linear','extrap'); % I1 in u
    dI = (I0 - Iu).*Is; % treibende Kraft
    v = -(mu/h^4)*D*u + dI; % Residuum
    A = (mu/h^4)*D + diag(Is.^2); % Residuum-Ableitung
    du = A\v; % Newton Schritt
    u = u + omega*du; % gedaempfte Abschaetzung
    it = it+1;
    plot(x,I0,'b',x,I1,'g',x,Iu,'r'); % grafisch darstellen
    drawnow;
    pause(0.01);
end

% interpoliertes Ergebnis grafisch darstellen
nb = n;
for i=1:nb
    z = (i-1)/(nb-1); % interpolierte Funktion
    Iz = interp1(x,I1,x+z*(u-x),'linear','extrap')';
    plot(x,I0,'b',x,I1,'g',x,Iz,'r'); % grafisch darstellen
    pause(0.01);
end
```


Werkzeuge für dynamische Systeme: NETLOGO und VENSIM

<http://ccl.northwestern.edu/netlogo/>



<http://www.vensim.com/>

