02-10-2010

Image processing by deformation — Public lecture

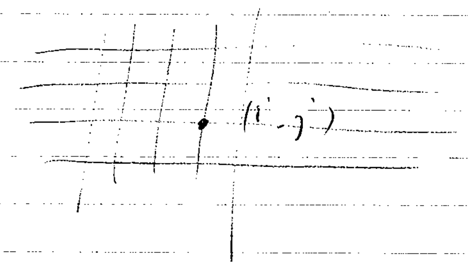
First talk: 1) how to view Gaussian filter as a deformation process;

2) Introducing level set method.

Example 1: Gaussian denoise: $u(x, y, t)$

$$
(1) \quad \begin{cases}
u_t = \Delta u = u_{xx} + u_{yy} & \Omega \times (0, \infty) \\
u(x, y, 0) = I(x, y) & \Omega \\
\dfrac{\partial u}{\partial \nu} = \nabla u \cdot \nu = 0 & \partial\Omega
\end{cases}
$$

Grid of an image



m moment

$$u^m(i, j) := u(i, j, m)$$

Then
$$u_{xx}(i, j, n) = \frac{u^n(i+1, j) + u^n(i-1, j) - 2u^n(i, j)}{\Delta x^2}$$

$$= \frac{u^n_{i+1,j} + u^n_{i-1,j} - 2u^n_{i,j}}{\Delta x^2}$$

$$u_{yy}(i, j, n) = \frac{u^n_{i,j+1} + u^n_{i,j-1} - 2u^n_{i,j}}{\Delta y^2}$$

$$u_t(i, j, n) = \frac{u(i, j, n+1) - u(i, j, n)}{\Delta t} = \frac{u^{n+1}_{i,j} - u^n_{i,j}}{\Delta t}$$

02-10-2010

Publ...

$$\Rightarrow \quad u_{i,j}^{n+1} = \frac{\Delta t}{\Delta x^2}\left[u_{i+1,j}^{n} + u_{i-1,j}^{n} + u_{i,j+1}^{n} + u_{i,j-1}^{n} - 4u_{i,j}^{n}\right] + u_{i,j}^{n}$$

Or. $u_{i,j}^{n+1} = u_{i,j}^{n} + \text{step} \cdot \Delta u.$

Why it works? Where does the equation comes from? (Diffusion)

Consider functional

$$F(u) := \int_\Omega |\nabla u|^2 \, dx\,dy = \int_\Omega (u_x^2 + u_y^2)\, dx\,dy.$$

Find a path to decrease the value of $F(u)$.

Recall: from Calculus course:

$z = f(x,y) : \mathbb{R}^2 \to \mathbb{R}$, $\nu = (\nu_1, \nu_2)$ unit vector

Change of f along ν direction (Directional derivative)

$$\partial_\nu f := \nabla f \cdot \nu. \quad \text{So if } \nu = \begin{cases} \dfrac{\nabla f}{|\nabla f|} \\ \text{or} \\ -\dfrac{\nabla f}{|\nabla f|} \quad \text{negative} \end{cases}$$

f is decreasing or increasing fastest.

A bit complicated for functional:

02-10-2010

Page 3.

Public lecture

For a given $I(x,y)$ (Assume it is smooth)

Consider a family of parametrized function $u(x,y,t)$ s.t

$u(x,y,0) = I(x,y)$

Then $F(u) = g(t)$ a function of t.

$$\partial_t F(u) = \partial_t \int_\Omega |\nabla u|^2 \, dx\,dy = 2\int_\Omega \langle \nabla u, \nabla u_t \rangle \, dx\,dy$$

$$\underset{\text{formula}}{\overset{\text{Green's}}{=\!=}} -\int_\Omega \langle \Delta u, u_t \rangle \, dx\,dy + \int_{\partial\Omega} \frac{\partial u}{\partial \nu} \cdot u_t \, ds$$

(Green's formula : $\int_\Omega u \cdot \Delta v \, dx\,dy = \int_{\partial\Omega} u \cdot \frac{\partial v}{\partial \nu} \, ds - \int_\Omega \nabla u \cdot \nabla v \, dx\,dy$)

So, if we choose

$$\begin{cases} u_t = \Delta u \\ \frac{\partial u}{\partial \nu} = 0 \end{cases}$$

then $\partial_t F(u) \downarrow$.

Experiment with Matlab.

Example 2 : Curve flow and level set method.

Background: Another way to average or smooth out.

" Deforming a simple curve into a circle "

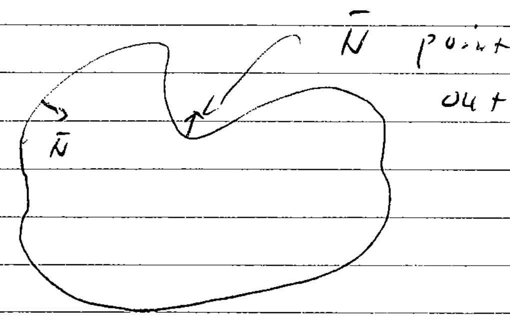
or an ellipse.

Curve flow problem (1985).

plane curve $\bar{F}_0(p) := (x(p), y(p)) : p \in [0,1] \longrightarrow \mathbb{R}^2$

$\bar{N}(p)$: unit inner normal vector

(roughly = second derivative of $\bar{F}$)



Consider

$$(2) \quad \begin{cases} \bar{F}_t(p,t) = \tau \bar{N}(p,t) \\ \bar{F}(p,0) = \bar{F}_0 \end{cases}$$

rounder & rounder.

① $\tau = k(p,t) \longleftarrow$ curve shortening flow

② $\tau = k^{1/3}(p,t) \longleftarrow$ Affine flow.

Level set method: Curve flow for denoising.

Consider a gray level image

$$\phi(x,y) : \Omega \longrightarrow \mathbb{R}^+$$

For given constant $c \geq 0$, then are contour curve(s).

$$\phi(x,y) = c.$$

Now, we will move all contour curves simultaneously:

$$\phi[x(t), y(t)] = c$$

Then
$$\phi_x \cdot x_t + \phi_y \cdot y_t + \phi_t = 0.$$

i.e. we want
$$\phi_t = -(x_t, y_t)(\phi_x, \phi_y).$$

If all contour curves are moved by curve flow (2):
$$(x_t, y_t) = \tau \vec{N} = -\tau \cdot \left(\frac{\phi_x}{|\nabla\phi|}, \frac{\phi_y}{|\nabla\phi|} \right)$$

Then
$$\phi_t = \tau \cdot |\nabla\phi| \;!$$

① $\tau = k$,
$$\phi_t = \frac{\phi_x^2 \phi_{yy} + \phi_y^2 \phi_{xx} - 2\phi_x \phi_y \phi_{xy}}{|\nabla\phi|^2}$$

② $\tau = k^{1/3}$,
$$\phi_t = (\phi_x^2 \phi_{yy} + \phi_y^2 \phi_{xx} - 2\phi_x \phi_y \phi_{xy})^{1/3}.$$

Since
$$k = \frac{\phi_x^2 \phi_{yy} + \phi_y^2 \phi_{xx} - 2\phi_x \phi_y \phi_{xy}}{|\nabla\phi|^3}.$$

Experiments

```
function y=gaussian(F, items, step)

%%Gaussian flow to smooth or blur image. MJ 7-05-2009.

[m,n]=size(F);
a=zeros(m,n);
b=zeros(m,n);
c=zeros(m,n);
d=zeros(m,n);

for k=1:items;

    f=F;
    a(1:m-1,:)=f(2:m,:);    %a(i,j)=f(i+1,j)
    a(m,:)=f(m,:);
    b(2:m,:)=f(1:m-1,:);%b(i,j)=f(i-1,j)
    b(1,:)=f(1,:);
    c(:,1:(n-1))=f(:,2:n);%c(i,j)=f(i,j+1)
    c(:,n)=f(:,n);
    d(:,2:n)=f(:,1:n-1);%d(i,j)=f(i,j-1)
    d(:,1)=f(:,1);



    fxx=a+b-2.*f; %fxx(i,j)=f(i+1,j)+f(i-1,j)-2f(i,j)
    fyy=c+d-2.*f; %fyy(i,j)=f(i,j+1)+f(i,j-1)-2f(i,j)



    ft=fxx+fyy;

       F=F+step.*ft; %stability: step<1/4.

end;

y=F;
```

```
function y=affine(f)
[m,n]=size(f);
a=zeros(m,n);
b=zeros(m,n);
c=zeros(m,n);
d=zeros(m,n);

for k=1:10;

    a(1:m-1,:)=f(2:m,:);    %a(i,j)=f(i+1,j)
    a(m,:)=f(m,:);
    b(2:m,:)=f(1:m-1,:);%b(i,j)=f(i-1,j)
    b(1,:)=f(1,:);
    c(:,1:(n-1))=f(:,2:n);%c(i,j)=f(i,j+1)
    c(:,n)=f(:,n);
    d(:,2:n)=f(:,1:n-1);%d(i,j)=f(i,j-1)
    d(:,1)=f(:,1);


    fx=a-f; %fx(i,j)=f(i+1,j)-f(i,j)
    fy=c-f; %fy(i,j)=f(i,j+1)-f(i,j)
    fxx=a+b-2.*f; %fxx(i,j)=f(i+1,j)+f(i-1,j)-2f(i,j)
    fyy=c+d-2.*f; %fyy(i,j)=f(i,j+1)+f(i,j-1)-2f(i,j)

    e(:,1:(n-1))=fx(:,2:n);%e(i,j)=fx(i,j+1)
    e(:,n)=fx(:,n);
    fxy=e-fx;%fxy(i,j)=(fx)y(i,j)=fx(i,j+1)-fx(i,j)

    %Euclidean shortening flow power=1
    %ft=(fx.^2.*fyy+fy.^2.*fxx-2.*fx.*fy.*fxy)./(fx.^2+fy.^2+0.00001);

    %affine flow power=1/3
    ft=nthroot((fy.^2.*fxx+fx.^2.*fyy-2.*fx.*fy.*fxy),3);


    f=f+0.1*ft;

end;
y=f;
```

For the first mathematical geoscience

Seminar