Vision, Image & Signal Processing (VISP)

HERIOT WATT UNIVERSITY

Implicit Filtering for Image and Shape Processing

Alex Belyaev

Electrical, Electronic & Computer Engineering

School of Engineering & Physical Sciences

Heriot-Watt University

Edinburgh

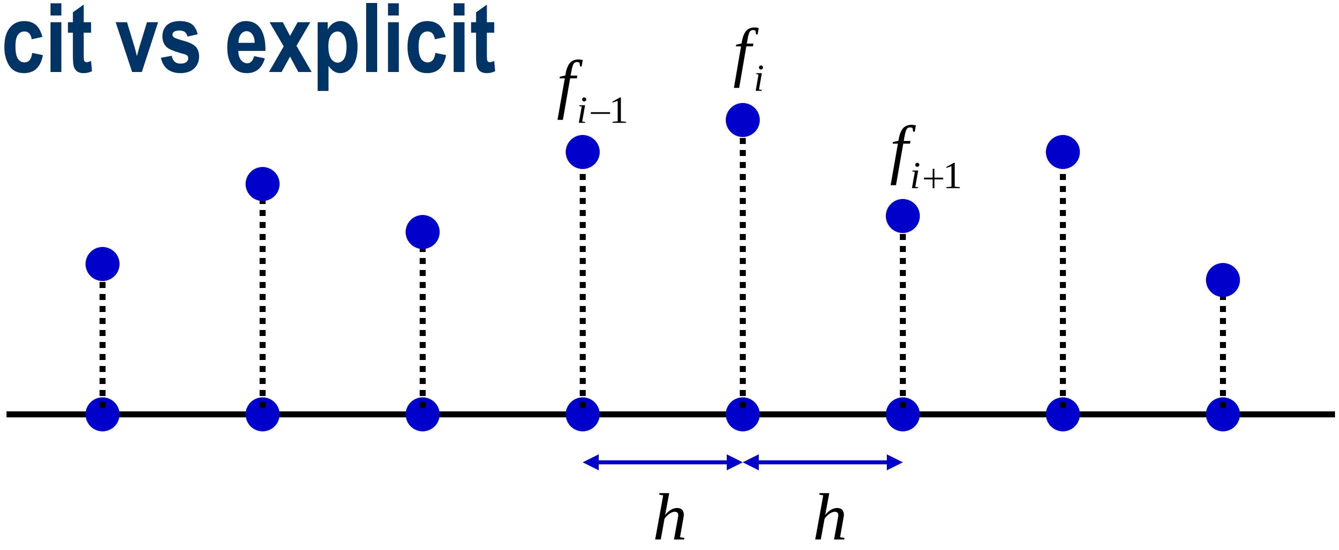
Implicit filtering and its applications

- A. Belyaev, “On implicit image derivatives and their applications.” BMVC 2011, Dundee, Scotland, UK, August 2011.

- A. Belyaev and H. Yamauchi, “Implicit filtering for image and shape processing.” VMV 2011, Berlin, Germany, October 2011.

- A. Belyaev, B. Khesin, and S. Tabachnikov, “Discrete speherical means of directional derivatives and Veronese maps.” *Journal of geometry and Physics*, 2011. Accepted.

Implicit vs explicit



Discrete signal sampled regularly with spacing h

$$f_i' = \frac{1}{2h}\left(f_{i+1} - f_{i-1}\right)$$

Standard **explict** finite difference scheme

$$\frac{1}{w+2}\left(f_{i-1}' + w\, f_i' + f_{i+1}'\right) = \frac{1}{2h}\left(f_{i+1} - f_{i-1}\right)$$

An **implicit** finite difference scheme

$$\frac{f(x+h) - f(x-h)}{2h} = f'(x) + \frac{h^2}{6}\frac{d^2}{dx^2} f'(x) + O(h^4)$$

$$= f'(x) + \frac{h^2}{6}\frac{1}{h^2}\left[f'(x+h) - 2f'(x) + f'(x-h)\right] + O(h^4)$$

$$= \frac{1}{6}\left[f'(x+h) + 4f'(x) + f'(x-h)\right] + O(h^4), \quad h \ll 1$$

w=4 gives a higher approximation order for small h.

A DSP approach to estimating Image Derivatives

$$L\left[e^{j\omega x}\right] \longleftrightarrow H \quad \omega \qquad \text{frequency response, } j=\sqrt{-1}$$

$$h=1 \quad \Rightarrow \quad -\pi<\omega<\pi$$

$$\frac{d}{dx}\left[e^{j\omega x}\right] = j\omega\, e^{j\omega x} \qquad \frac{d}{dx} \longleftrightarrow j\omega$$

$$\frac{f \quad x+1 \ - f \quad x-1}{2} \longleftrightarrow j\sin\omega$$

Delivers a good approximation of $j\omega$ for small ω only

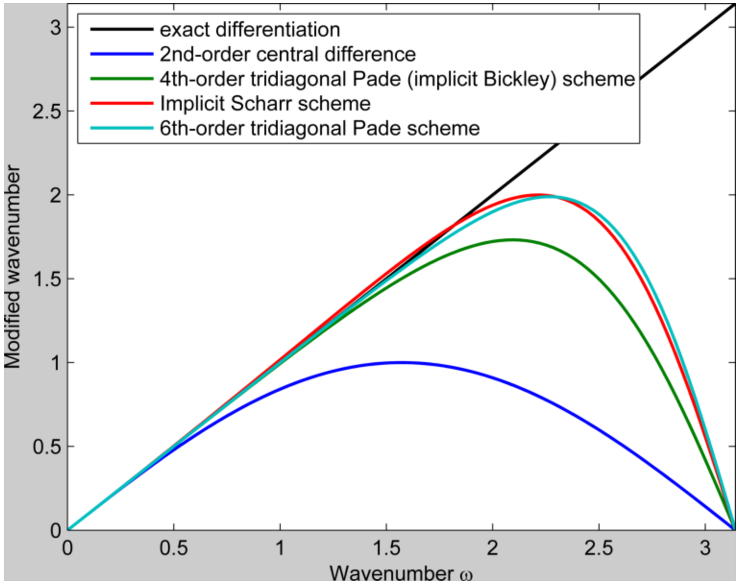
Implicit finite differences: an example

$$f' \quad x-h \ +4f' \quad x \ +f' \quad x+h \ = \frac{3}{h}\left[f \quad x+h \ - f \quad x-h \ \right] + O \quad h^4$$

$$H \quad \omega \ = j\frac{3\sin\omega}{2+\cos\omega}$$

- Non-causal IIR filters in the DSP language
- Rational (Pade) approximations in Maths

A DSP approach to estimating Image Derivatives



$$\frac{1}{w+2}\ \ f'_{i-1}+w\,f'_i+f'_{i+1}\ \ =\frac{1}{2h}\ \ f_{i+1}-f_{i-1}$$

$$H\ \ \omega\ \ = j\sin\omega\frac{w+2}{w+2\cos\omega}$$

Let us introduce

- ***implicit Scharr scheme*** $w=10/3$
- ***implicit Bickley scheme*** $w=4$

A 6-order Pade scheme:

$$f'\ \ x-h\ \ +3f'\ \ x\ \ +f'\ \ x+h \qquad H\ \ \omega\ \ = j\frac{\sin\omega\ \ 28+2\cos\omega}{6\ \ 3+2\cos\omega}$$

$$=\frac{1}{12h}\left[f\ \ x+2h\ \ +28f\ \ x+h\ \ -28f\ \ x-h\ \ -f\ \ x-2h\ \ \right]+O\ \ h^6$$

Commonly used discrete gradients & Laplacians

Rotation-invariant differential quantities (operators) used widely in Image Processing and Computer Vision:

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}, \quad \nabla = \begin{bmatrix} \partial/\partial x \\ \partial/\partial y \end{bmatrix}, \quad \left(\frac{\partial}{\partial x}\right)^2 + \left(\frac{\partial}{\partial y}\right)^2, \quad \left(\frac{\partial^2}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2}{\partial y^2}\right)^2$$

Need for accurate discrete approximations. The standard discrete approximations are not sufficiently accurate.

$$\frac{\partial}{\partial x} \approx \frac{1}{2h \;\; w+2} \begin{bmatrix} -1 & 0 & 1 \\ -w & 0 & w \\ -1 & 0 & 1 \end{bmatrix}$$

$w = 1$ (Prewitt, 1970)

$w = 2$ (Sobel, 1970)

$w = 10/3$ (Scharr, 2000)

$w = 4$ (Bickley, 1947)

$w = \infty$ simple symmetric f.d.

$$\Delta \approx \frac{1}{h^2 \;\; w+2} \begin{bmatrix} 1 & w & 1 \\ w & -4 \;\; w+1 & w \\ 1 & w & 1 \end{bmatrix}$$

$w = 1$ (Gonzalez &Woods)

$w = 2$ (Kamgar-Parsi & Resenfeld, 1999)

$w = 4$ Mehrstellen Laplacian

$w = \infty$ standard 5-point stencil

Approximation Accuracy and Rotational Invariance

$$\frac{1}{2h}\begin{bmatrix}0&0&0\\-1&0&1\\0&0&0\end{bmatrix}=\frac{\partial}{\partial x}+O\;h^{2}$$

$$\frac{1}{12h}\begin{bmatrix}-1&0&1\\-4&0&4\\-1&0&1\end{bmatrix}=\left(1+\frac{h^{2}}{12}\Delta\right)\frac{\partial}{\partial x}+O\;h^{4}$$

Optimally rotation-invatiant?

$$\frac{1}{h^{2}}\begin{bmatrix}0&1&0\\1&-4&1\\0&1&0\end{bmatrix}=\Delta+\frac{h^{2}}{12}\left(\frac{\partial^{4}}{\partial x^{4}}+\frac{\partial^{4}}{\partial y^{4}}\right)+O\;h^{4}$$

$$\frac{1}{6h^{2}}\begin{bmatrix}1&4&1\\4&-20&4\\1&4&1\end{bmatrix}=\Delta+\frac{h^{2}}{12}\Delta^{2}+O\;h^{4}$$

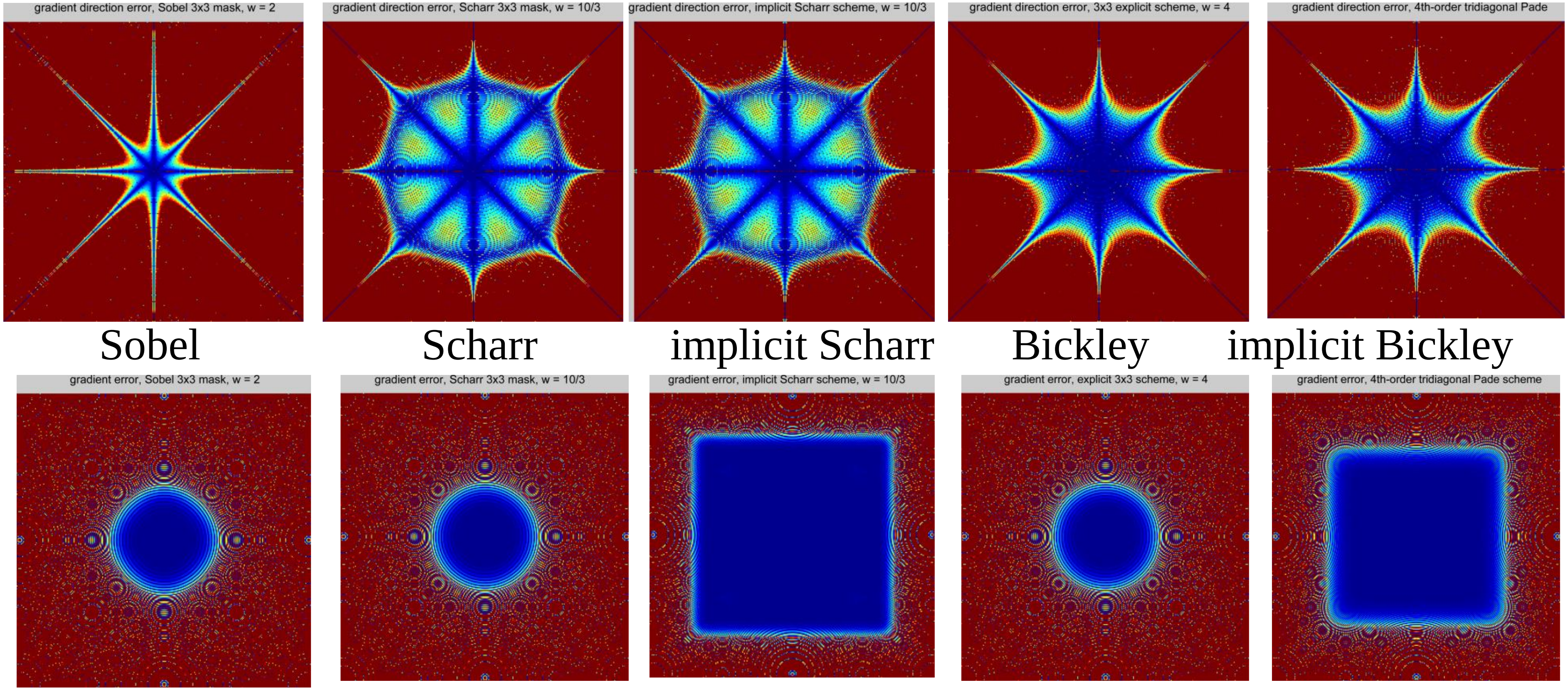
(Horn, *Robot Vision*)

Two natural questions to ask:

- Is it possible to achieve a better approximation accuray for the same computational cost?
- Why shuld we assume that the grid spacing (pixel size) *h* tends to 0?

Estimating the gradient direction and maginude

gradient direction error



Sobel Scharr implicit Scharr Bickley implicit Bickley

gradient magnitude error

Explicit schemes and their implicit counterparts deliver remarkably similar estimates of the gradient direction field.

Explicit vs. Implicit

$$\frac{\partial}{\partial x} \approx \frac{1}{2h \;\; w+2}\begin{bmatrix} -1 & 0 & 1 \\ -w & 0 & w \\ -1 & 0 & 1 \end{bmatrix}$$

Smoothing introduced by [-1 0 1]/2 in *x*-direction is compensated by applying [1 w 1]/(w+2) smoothing in *y*-direction

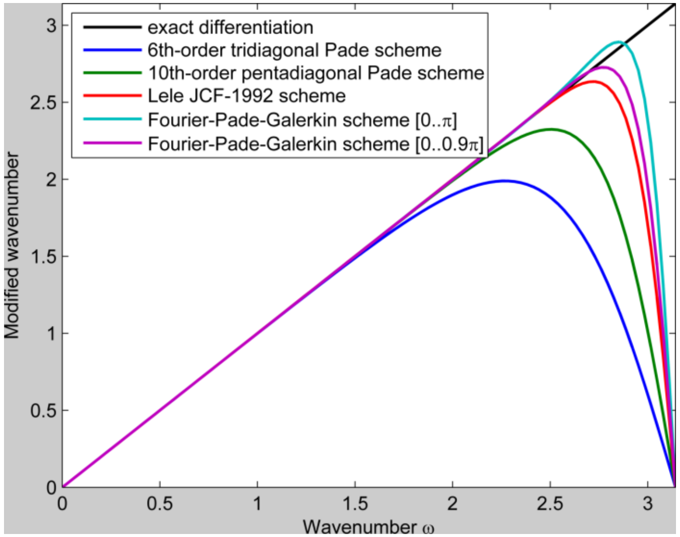
$$\frac{1}{w+2} \quad f'_{i-1} + w\, f'_{i+1} + f'_{i+1}$$

$$= \frac{1}{2h} \quad f_{i+1} - f_{i-1}$$

Smoothing introduced by [-1 0 1]/2 in *x*-direction is comensated by applying [1 w 1]/(w+2) smoothing to the derivative.

Given an explicit scheme and its implicit counterpart, both the schemes produce similar estimates of the gradient direction, however the implicit scheme does a better job in estimating the gradient magnitude.

High-resolution schemes



S. K. Lele, “Compact finite difference schemes with spectral like resolution.” Journal of Computational Physics, 1992.

$$\beta f'_{i-2} + \alpha f'_{i-1} + f'_i + \alpha f'_{i+1} + \beta f'_{i+2}$$

$$= \frac{a}{2} \ \ f_{i+1} - f_{i-1} \ \ + \frac{b}{4} \ \ f_{i+2} - f_{i-2} \ \ + \frac{c}{6} \ \ f_{i+3} - f_{i-3}$$

$$H \ \ \omega \ \ = j \frac{a \sin \omega + \ \ b/2 \ \ \sin 2\omega + \ \ 3 \ \ \sin 3\omega}{1 + 2\alpha \cos \omega + 2\beta \cos 2\omega}$$

Lele scheme: $\alpha = 0.5771439, \quad \beta = 0.0896406$

$a = 1.302566, \quad b = 0.99355, \quad c = 0.03750245$

Fourier-Pade-Galerkin approximations 1

Space or trigonometric polynomials of degree N

$$\mathcal{F}_N = \text{span}\ \ e^{jn\omega} : -N \leq n \leq N$$

Rational Fourier series

$$R_{kl}\ \ \omega\ \ = P_k\ \ \omega\ \ / Q_l\ \ \omega$$

$$P_k\ \ \omega\ \ \in \mathcal{F}_k, \quad Q_l\ \ \omega\ \ \in \mathcal{F}_l$$

$$f\ \ \omega\ \ \equiv \omega \approx R_{kl}\ \ \omega\ \ , \quad \omega \in\ \ -\pi, \pi$$

$$\int_{-\pi}^{\pi} \left[Q_l\ \ \omega\ \ f\ \ \omega\ \ - P_k\ \ \omega\ \ \right] \overline{g\ \ \omega\ \ } W\ \ \omega\ \ d\omega = 0 \quad \forall g\ \ \omega\ \ \in \mathcal{F}_{k+l}$$

$W\ \ \omega$ is a properly chosen weighting function.

It gives a system of k+l lnear equations with k+l unknowns.

In our case, k=3 and l=2.

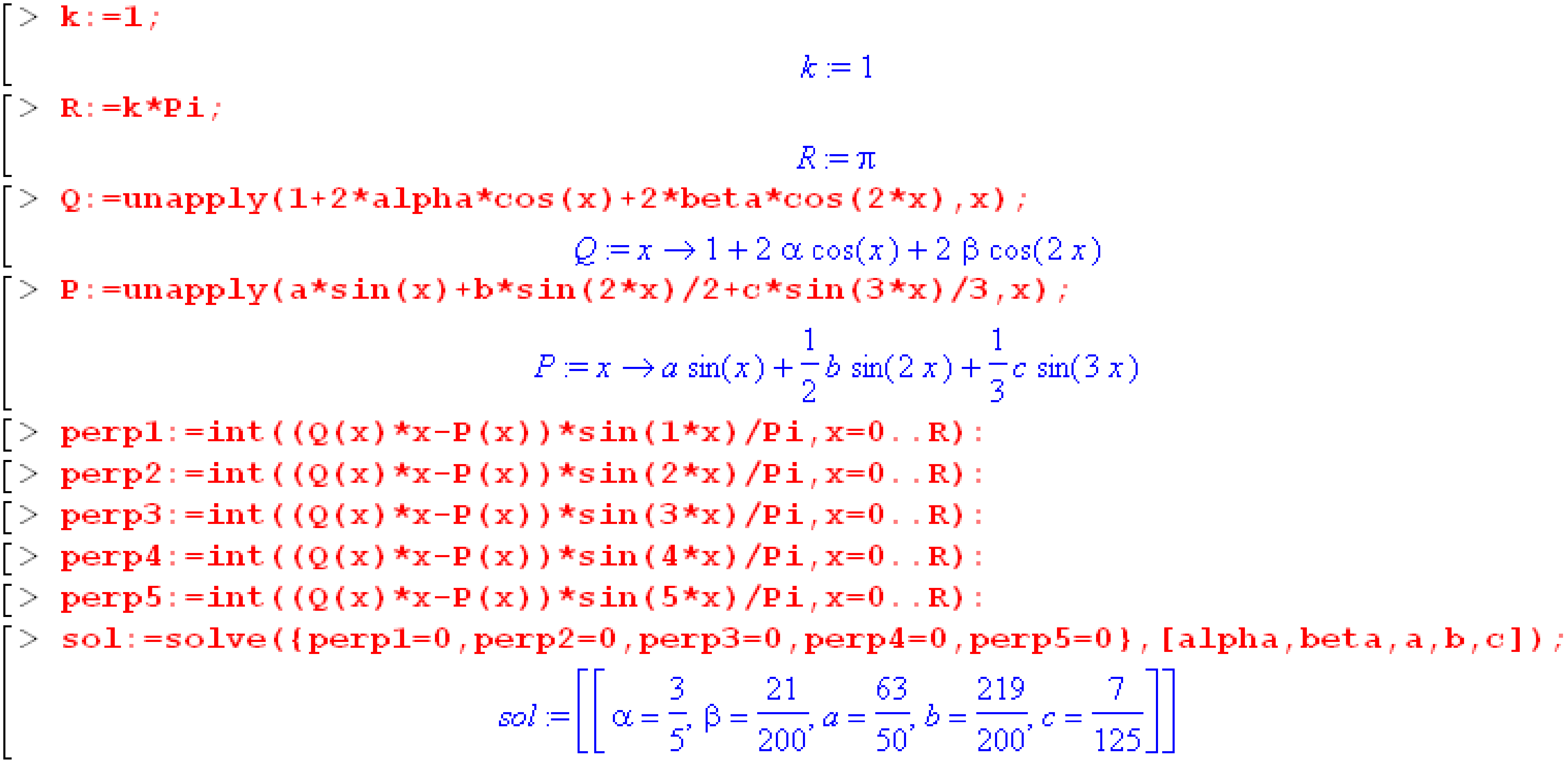
$$P_3\ \ \omega\ \ = a \sin\omega +\ \ b/2\ \ \sin 2\omega +\ \ 3\ \ \sin 3\omega$$

$$Q_2\ \ \omega\ \ = 1 + 2\alpha \cos\omega + 2\beta \cos 2\omega$$

Fourier-Pade-Galerkin approximations 2

A system of *k*+*l* linear equations with *k*+*l* unknowns. *k*=3 and *l*=2.

$$W \quad \omega \quad \equiv 1$$



```
> k:=1;
```

$$k := 1$$

```
> R:=k*Pi;
```

$$R := \pi$$

```
> Q:=unapply(1+2*alpha*cos(x)+2*beta*cos(2*x),x);
```

$$Q := x \rightarrow 1 + 2\,\alpha\cos(x) + 2\,\beta\cos(2\,x)$$

```
> P:=unapply(a*sin(x)+b*sin(2*x)/2+c*sin(3*x)/3,x);
```

$$P := x \rightarrow a\sin(x) + \frac{1}{2}\,b\sin(2\,x) + \frac{1}{3}\,c\sin(3\,x)$$

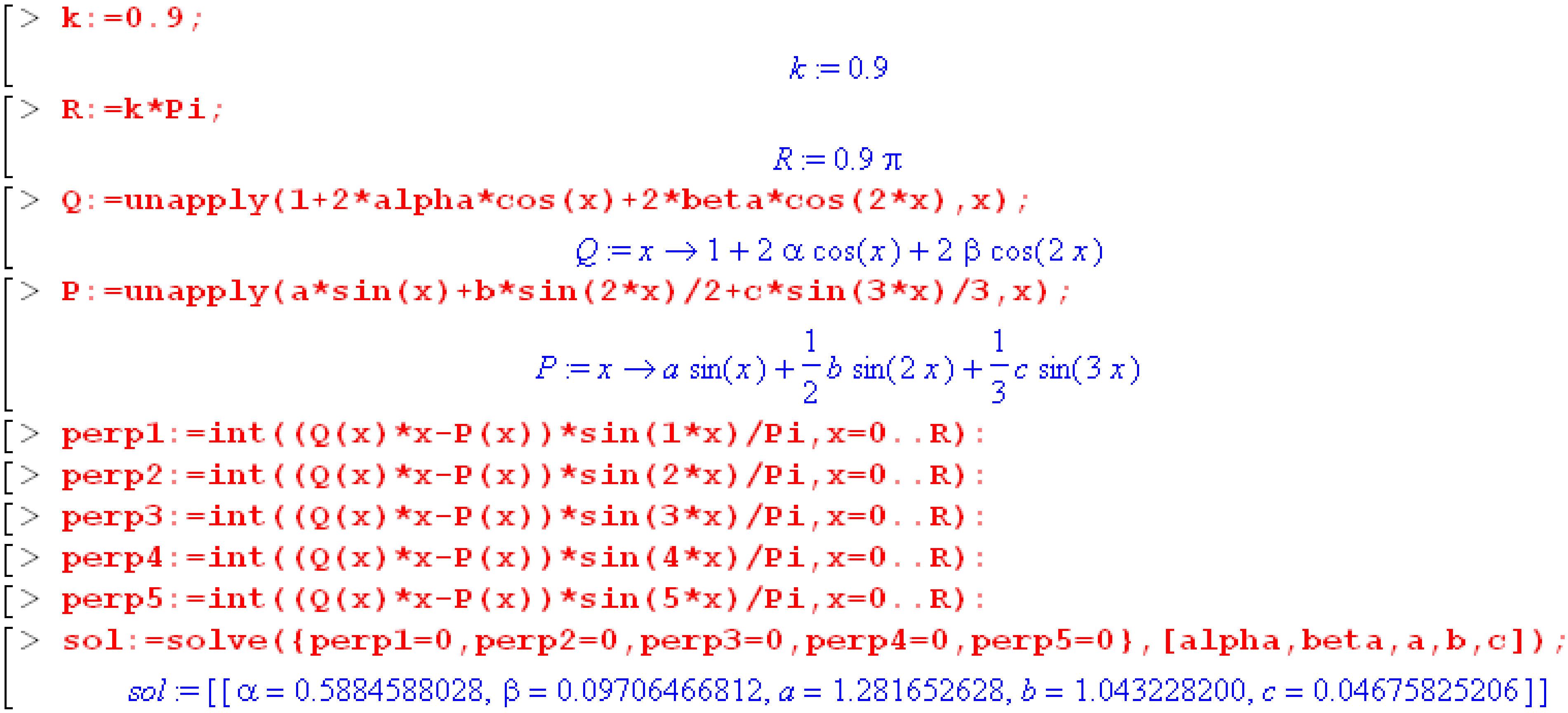
```
> perp1:=int((Q(x)*x-P(x))*sin(1*x)/Pi,x=0..R):
> perp2:=int((Q(x)*x-P(x))*sin(2*x)/Pi,x=0..R):
> perp3:=int((Q(x)*x-P(x))*sin(3*x)/Pi,x=0..R):
> perp4:=int((Q(x)*x-P(x))*sin(4*x)/Pi,x=0..R):
> perp5:=int((Q(x)*x-P(x))*sin(5*x)/Pi,x=0..R):
> sol:=solve({perp1=0,perp2=0,perp3=0,perp4=0,perp5=0},[alpha,beta,a,b,c]);
```

$$sol := \left[\left[\alpha = \frac{3}{5},\ \beta = \frac{21}{200},\ a = \frac{63}{50},\ b = \frac{219}{200},\ c = \frac{7}{125}\right]\right]$$

Fourier-Pade-Galerkin approximations 3

A system of k+l linear equations with k+l unknowns. k=3 and l=2.

$$W \quad \omega \quad = \begin{cases} 1 & 0 \le \omega \le 0.9\pi \\ 0 & 0.9\pi < \omega \le \pi \end{cases}$$



```
> k:=0.9;
```

$$k := 0.9$$

```
> R:=k*Pi;
```

$$R := 0.9\,\pi$$

```
> Q:=unapply(1+2*alpha*cos(x)+2*beta*cos(2*x),x);
```

$$Q := x \rightarrow 1 + 2\,\alpha\,\cos(x) + 2\,\beta\,\cos(2\,x)$$

```
> P:=unapply(a*sin(x)+b*sin(2*x)/2+c*sin(3*x)/3,x);
```

$$P := x \rightarrow a\,\sin(x) + \frac{1}{2}\,b\,\sin(2\,x) + \frac{1}{3}\,c\,\sin(3\,x)$$

```
> perp1:=int((Q(x)*x-P(x))*sin(1*x)/Pi,x=0..R):
> perp2:=int((Q(x)*x-P(x))*sin(2*x)/Pi,x=0..R):
> perp3:=int((Q(x)*x-P(x))*sin(3*x)/Pi,x=0..R):
> perp4:=int((Q(x)*x-P(x))*sin(4*x)/Pi,x=0..R):
> perp5:=int((Q(x)*x-P(x))*sin(5*x)/Pi,x=0..R):
> sol:=solve({perp1=0,perp2=0,perp3=0,perp4=0,perp5=0},[alpha,beta,a,b,c]);
```

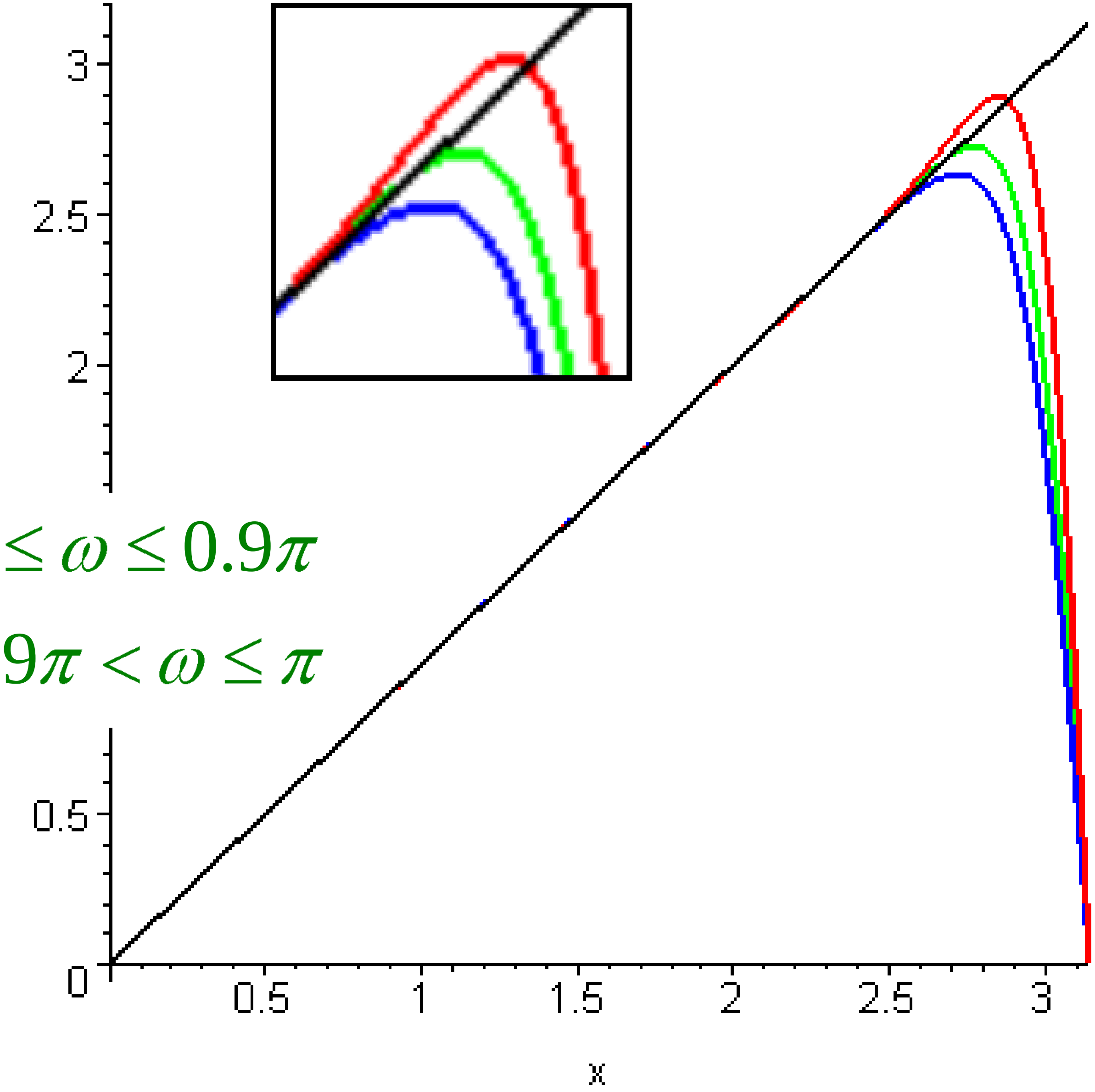
$$sol := [[\alpha = 0.5884588028,\ \beta = 0.09706466812,\ a = 1.281652628,\ b = 1.043228200,\ c = 0.04675825206]]$$

Fourier-Pade-Galerkin approximations 4

Lele scheme

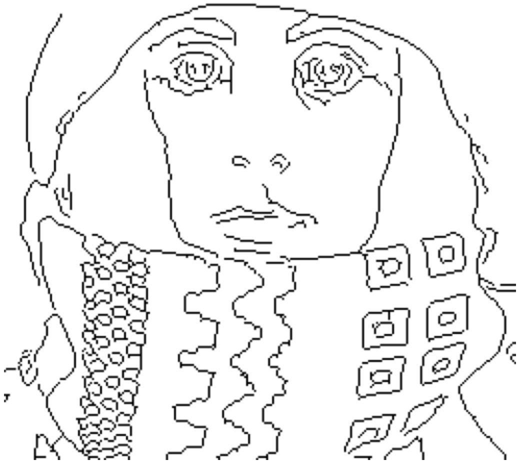
$$W \quad \omega \quad \equiv 1$$

$$W \quad \omega \quad = \begin{cases} 1 & 0 \le \omega \le 0.9\pi \\ 0 & 0.9\pi < \omega \le \pi \end{cases}$$

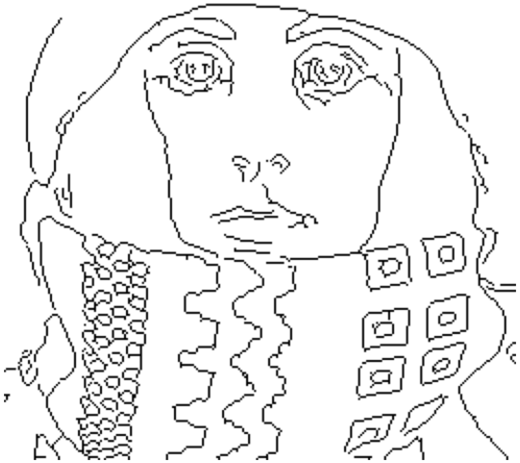


Applications: edge detection (Canny edge detection)

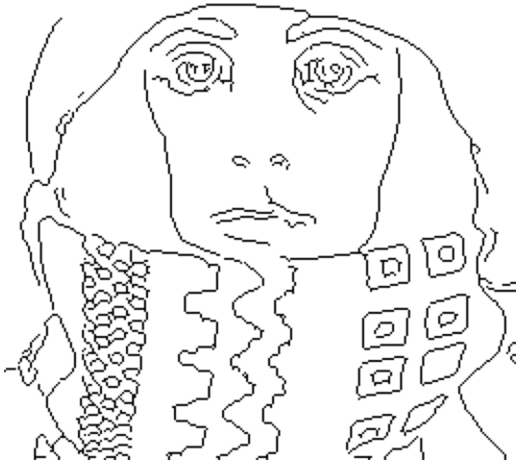
Sobel derivatives, σ = 0.75



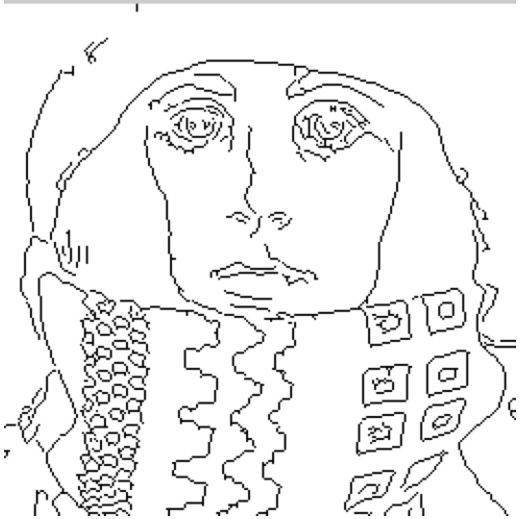
Explicit Scharr scheme, σ = 0.75



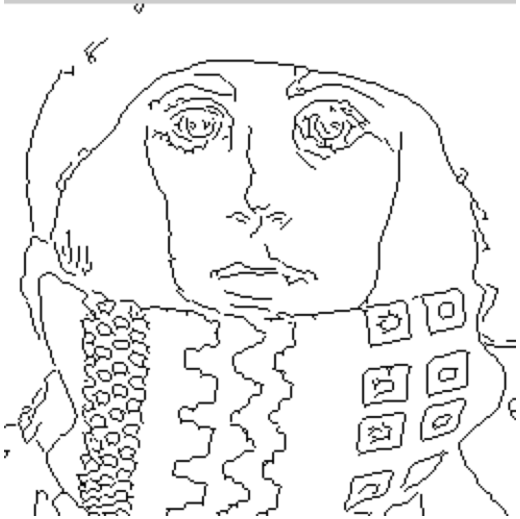
Farid-Simoncelli 5-tap filter, σ = 0.75



Implicit Scharr scheme, σ = 0.75



Fourier-Pade-Galerkin scheme, σ = 0.75

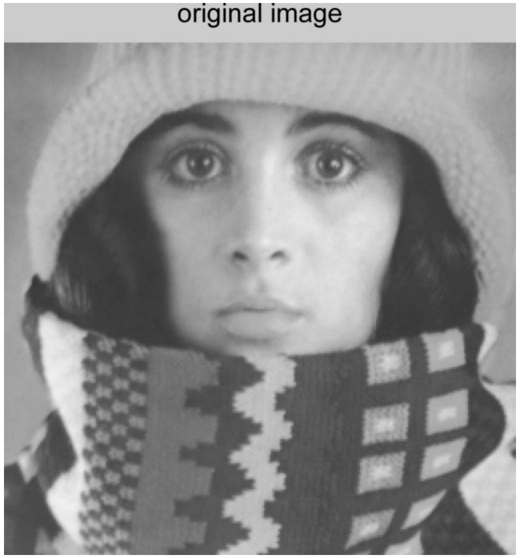


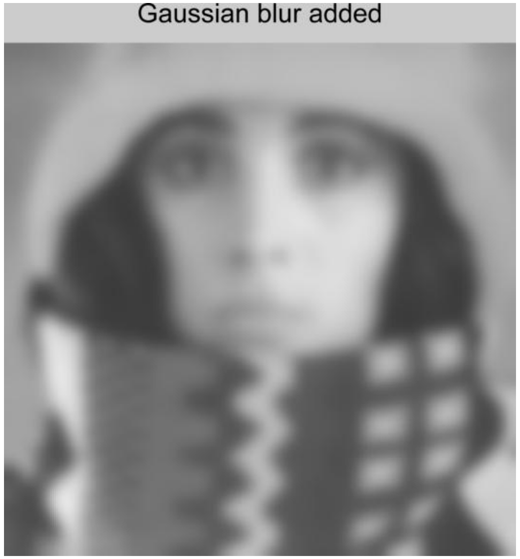
Applications: deblurring Gaussian blur

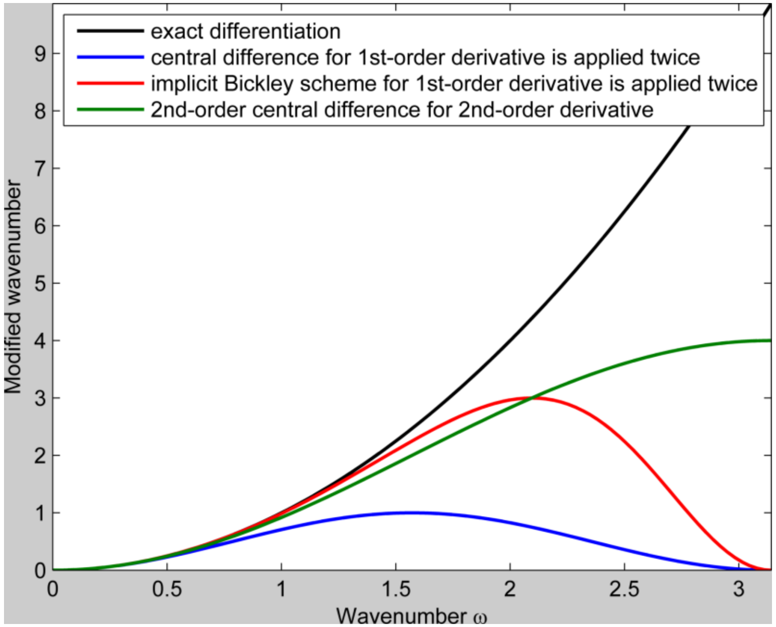
$$\frac{\partial}{\partial t} I \quad x, y, t \quad = -\Delta I \quad x, y, t$$

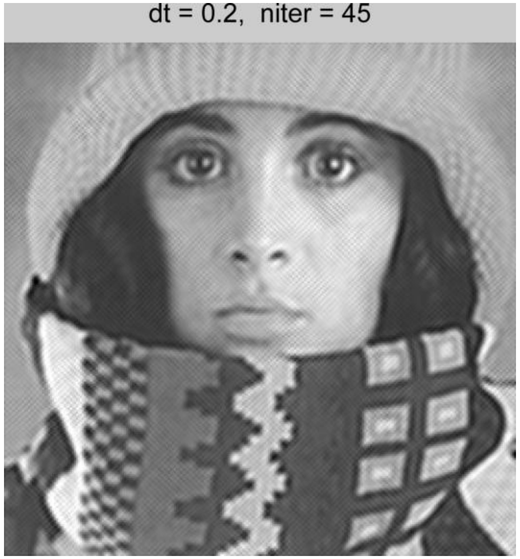
A higly unstable process.

The idea is to use a discrete Laplacian which dumps high frequences



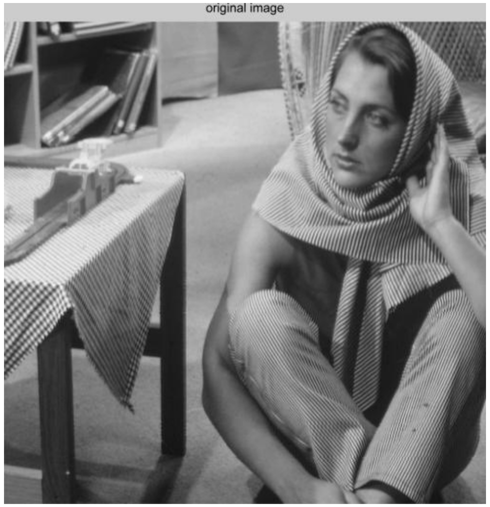


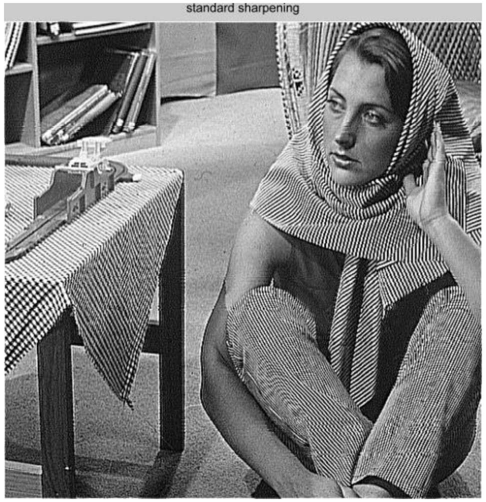


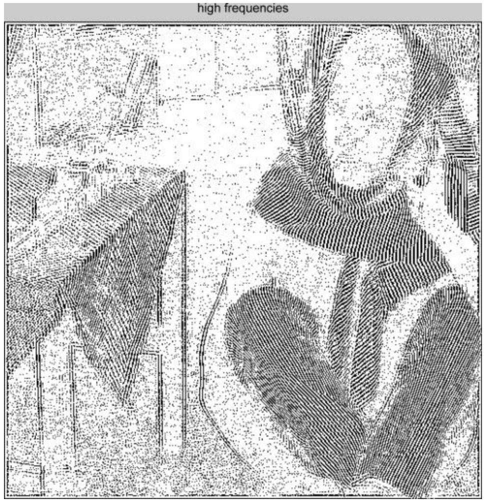


Restored image

Applications: unsharp masking

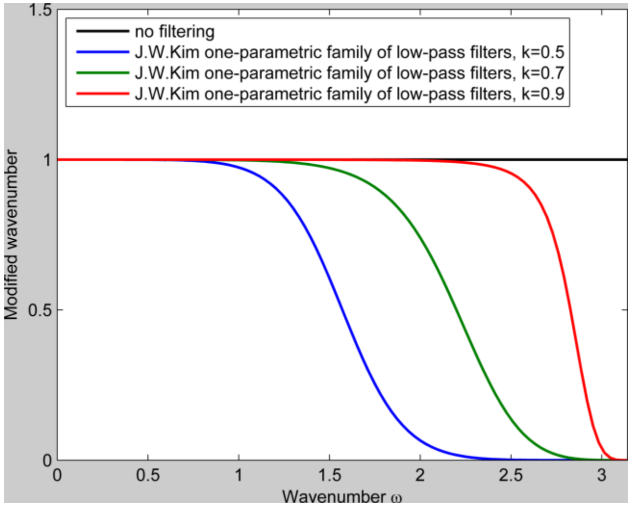




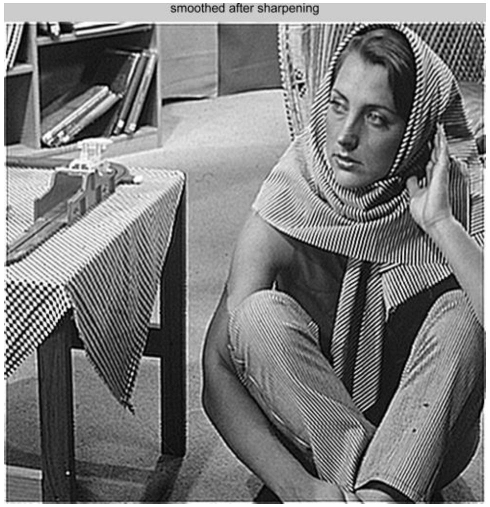


Standard unsharp masking oversharpens high-frequency details

$$I_{\text{sharp}}\quad x,y\;=I\quad x,y\;-\lambda\,\Delta I\quad x,y$$



Implicit filtereing does a good job in supressing oversharpened high-frequency details



Implicit filtering

$$H_{\varepsilon,p}\ \ \omega\ \ =\left(1+\varepsilon\tan^{2p}\frac{\omega}{2}\right)^{-1},\quad p=1,2,3,\ldots$$

frequency response function

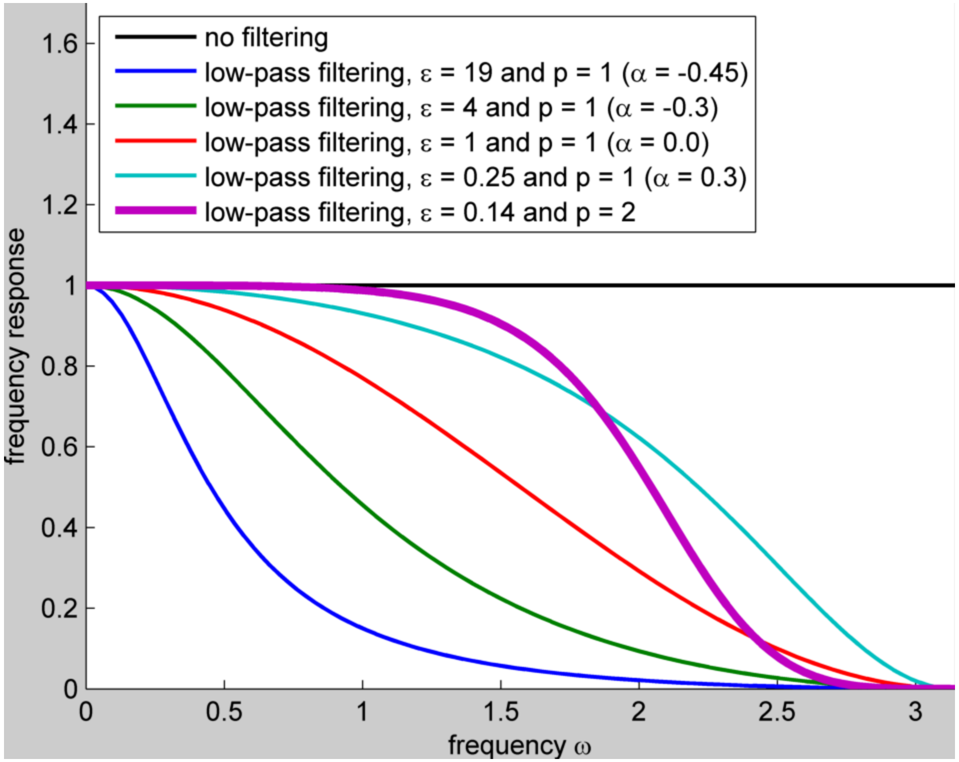
$$H_{\varepsilon,p}\ \ \omega\ \ =\begin{cases}1-\varepsilon\ \ \omega/2\ ^{2p}+O\ \ \omega^{2p+2} & \text{as}\ \ \omega\to 0\\ \dfrac{1}{\varepsilon}\left(\dfrac{\omega-\pi}{2}\right)^{2p}+O\ \ \ \omega-\pi\ ^{2p+2} & \text{as}\ \ \omega\to\pi\end{cases}$$

$$\frac{1}{1+2\alpha}\left[\alpha\hat{f}_{i-1}+\hat{f}_i+\alpha\hat{f}_{i+1}\right]$$

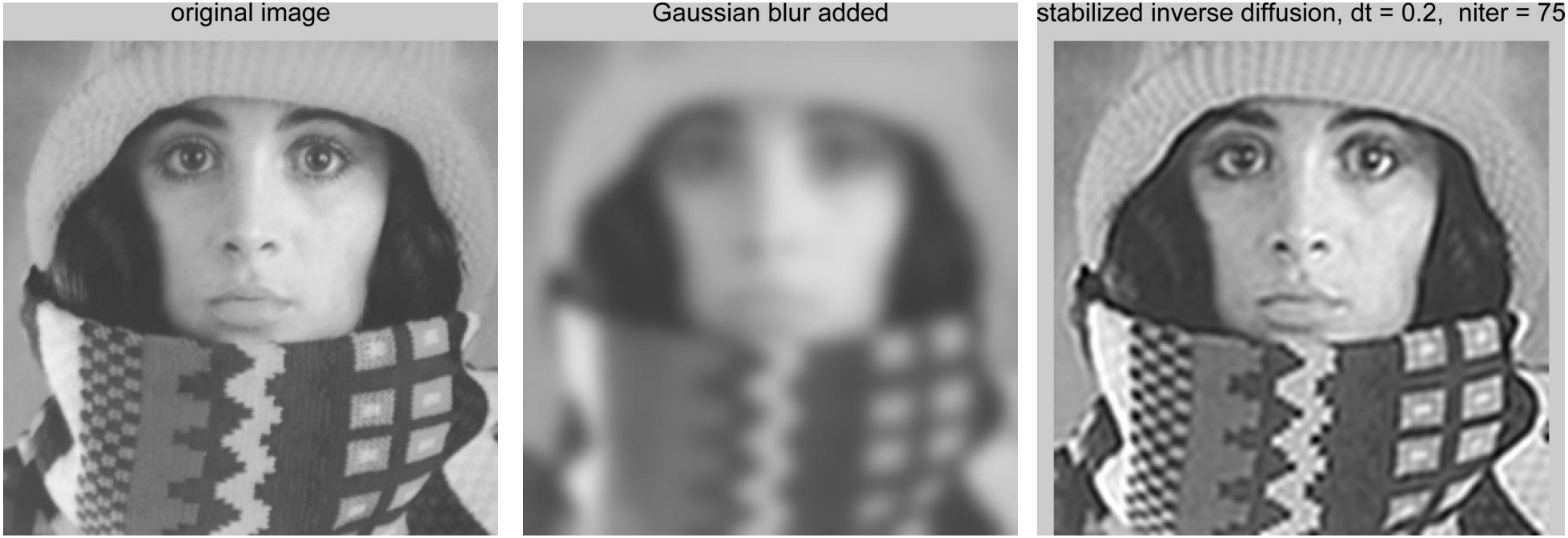
$$=\frac{1}{4}\ \ f_{i-1}+2f_i+f_{i+1}$$

$$H_\alpha\ \ \omega\ \ =\frac{1+2\alpha}{1+\alpha\cos\omega}\ \frac{1+\cos\omega}{2}$$

$$p=1,\quad \varepsilon=\frac{1-2\alpha}{1+2\alpha}$$



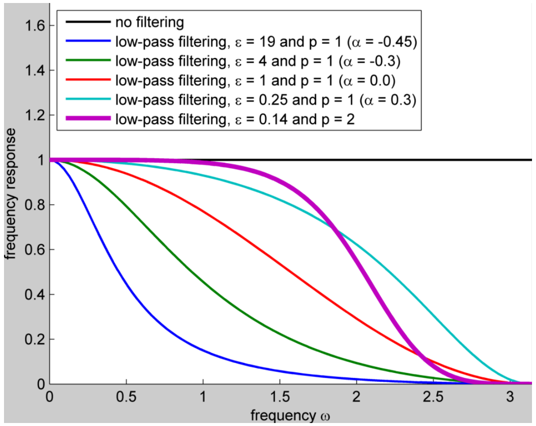
Stabilized inverse diffusion


original image

Gaussian blur added

stabilized inverse diffusion, dt = 0.2, niter = 75

$$I \quad x,y,t+dt \;=\; \text{low-pass}\left[I \quad x,y,t \; - dt\, \Delta_h I \quad x,y,t \; \right]$$



Implicit filtering and approximation subdivision

$$\frac{1}{1+2\alpha}\left[\alpha\,\hat{f}_{i-1}+\hat{f}_{i}+\alpha\,\hat{f}_{i+1}\right]=\frac{1}{4}\quad f_{i-1}+2f_{i}+f_{i+1}$$

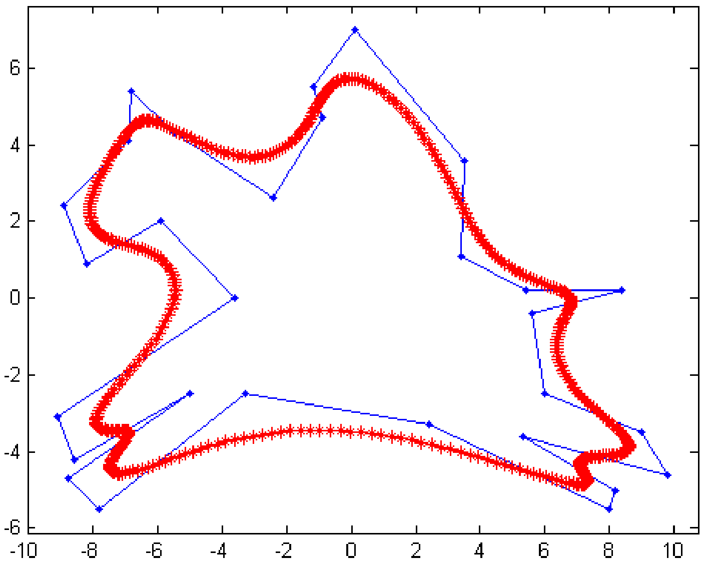
$$H_{\alpha}\quad\omega\quad=\frac{1+2\alpha}{1+\alpha\cos\omega}\,\frac{1+\cos\omega}{2}$$

$$\boldsymbol{u}_{2i}^{k}\ =\boldsymbol{v}_{i}^{k-1}\ ,\qquad \boldsymbol{u}_{2i+1}^{k}=\frac{1}{2}\ \ \boldsymbol{v}_{i}^{k-1}\ +\boldsymbol{v}_{i+1}^{k-1}$$

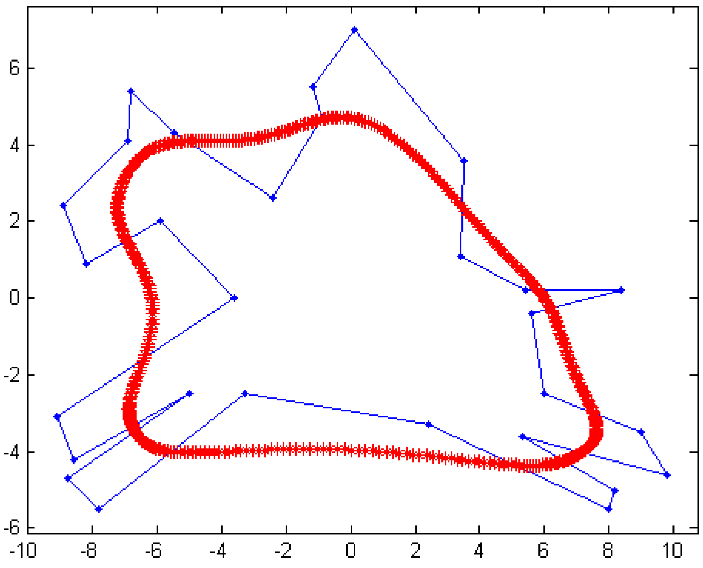
$$\frac{1}{1+2\alpha}\left[\alpha\boldsymbol{v}_{i-1}^{k}+\boldsymbol{v}_{i}^{k}\ +\alpha\boldsymbol{v}_{i+1}^{k}\right]=\frac{1}{2}\boldsymbol{u}_{i}^{k}\ +\frac{1}{4}\ \ \boldsymbol{u}_{i-1}^{k}+\boldsymbol{u}_{i+1}^{k}$$

Curve subdivision

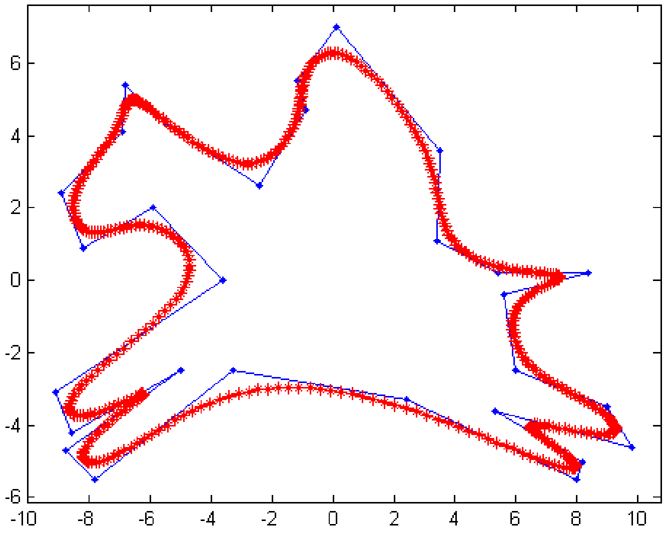
implicit subdivision: 4 iterations with $\alpha = -0.3$



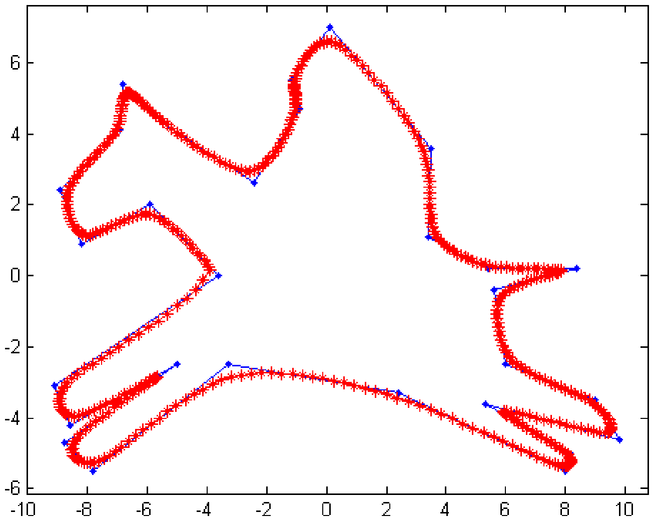
implicit subdivision: 4 iterations with $\alpha = -0.45$



implicit subdivision: 4 iterations with $\alpha = 0$



implicit subdivision: 4 iterations with $\alpha = 0.3$



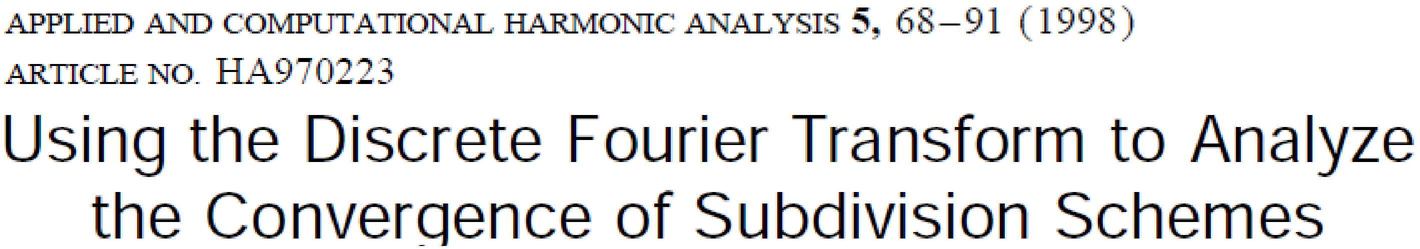
Implicit filtering and interpolatory subdivision

$$\alpha \hat{f}_{i-1} + \hat{f}_i + \alpha \hat{f}_{i+1} = \frac{a}{2} \quad f_{i-1/2} + f_{i+1/2} \quad + \frac{b}{2} \quad f_{i-3/2} + f_{i+3/2}$$

$$H \quad \omega \quad = \frac{a \cos \quad \omega/2 \quad + b \cos \quad 3\omega/2}{1 + 2\alpha \cos \omega}$$

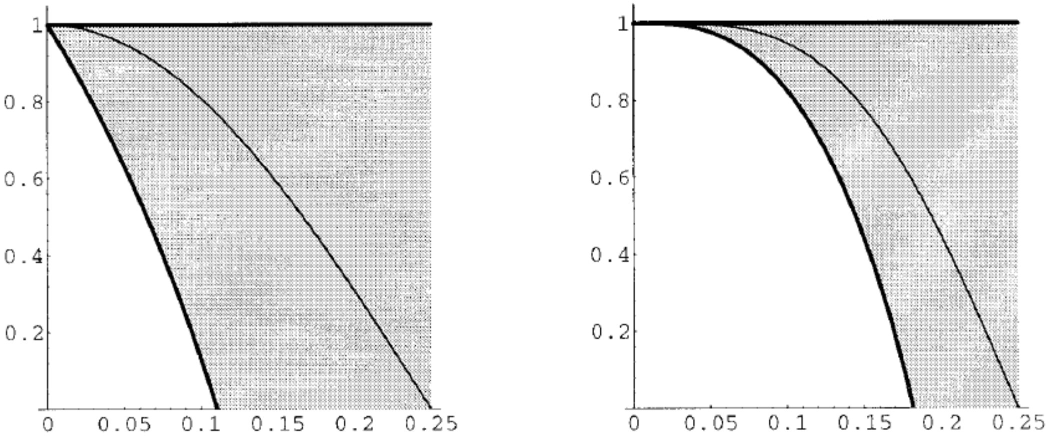
Dyn-Levin-Gregory: α=0, *a*=1/16, *b*=-1/9

Kobbelt K2 variational subdivision scheme: α=1/6, *a*=4/3, *b*=0

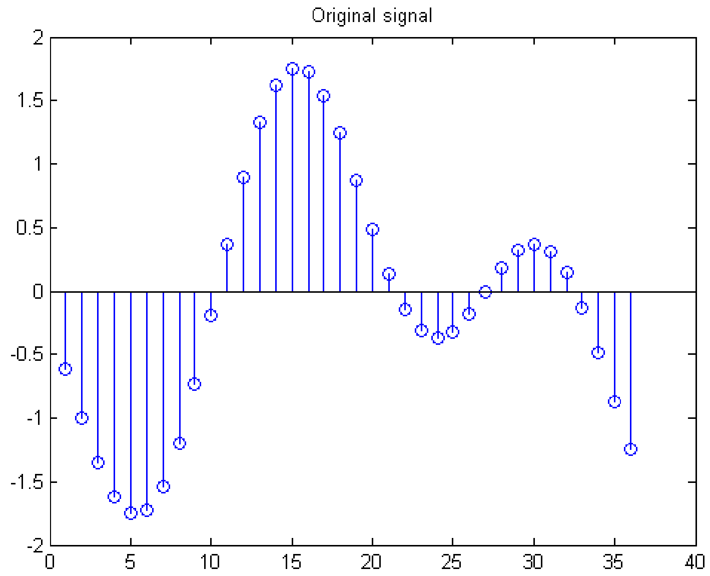


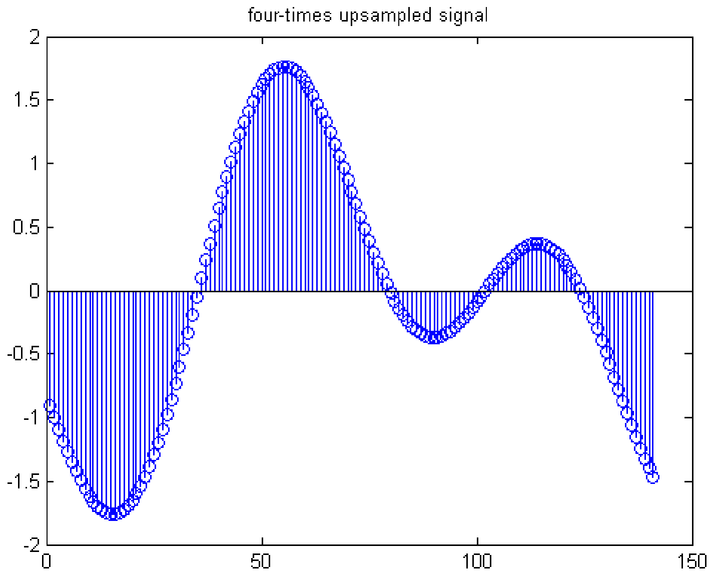
APPLIED AND COMPUTATIONAL HARMONIC ANALYSIS **5**, 68–91 (1998)
ARTICLE NO. HA970223

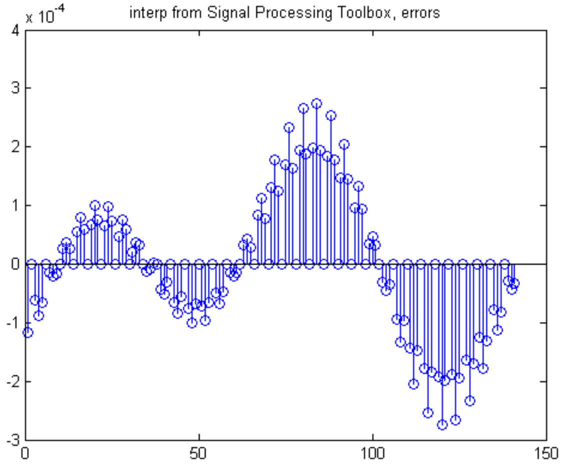
Using the Discrete Fourier Transform to Analyze the Convergence of Subdivision Schemes

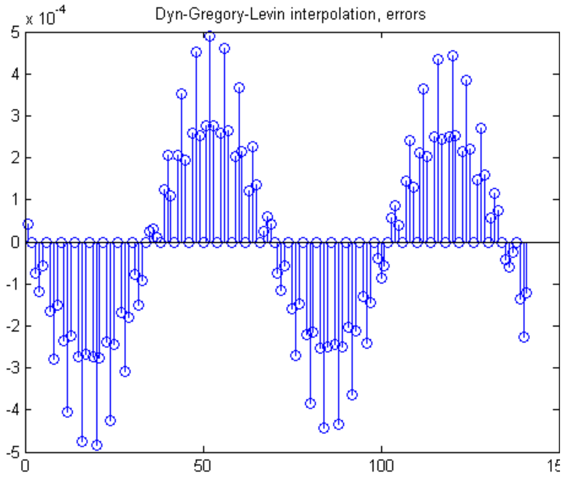


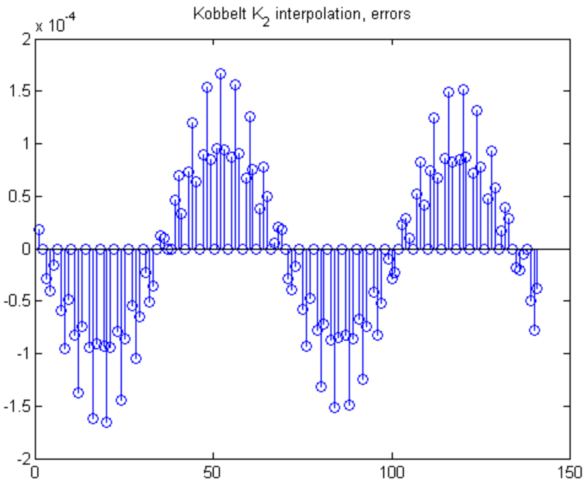
Implicit filtering and interpolatory subdivision





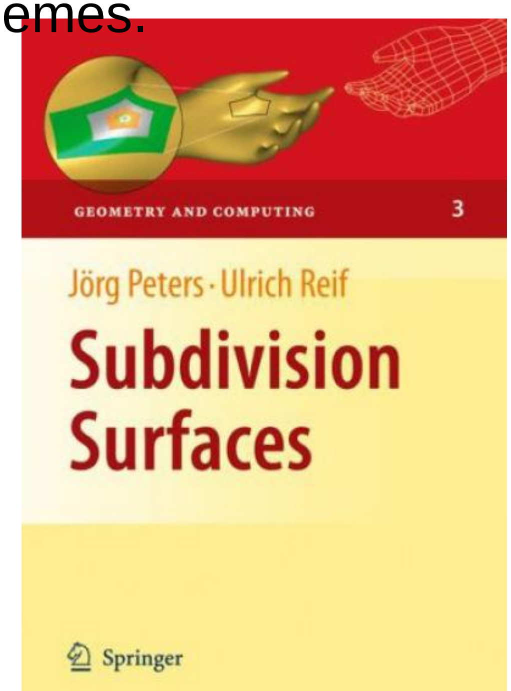






Implicit subdivision

- Implicit subdivision schemes were introduced by Kobbelt [1996,1998] in the case of interpolatory subdivision from a variational standpoint.
- Sabin [2010] does not mention them at all in his book (althought he cited that paper of Kobbelt).
- Peters and Reif [2008] devoted to variational subdivision only two sentences where the authors acknowledged its existence but wrongly stated that more or less nothing was known about the underlying theoretical properties of variational subdivision schemes.

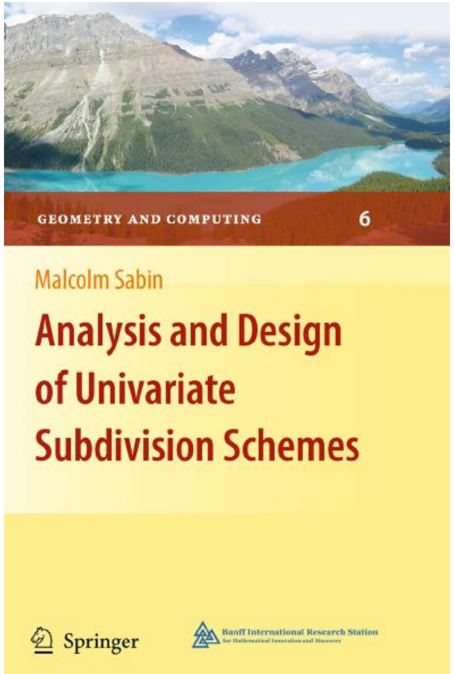


GEOMETRY AND COMPUTING 3

Jörg Peters · Ulrich Reif

Subdivision Surfaces

Springer



GEOMETRY AND COMPUTING 6

Malcolm Sabin

Analysis and Design of Univariate Subdivision Schemes

Springer

Banff International Research Station
for Mathematical Innovation and Discovery

Future research

- Weighted (non-iniform) implicit filtering schems → edge-aware image filtering (in a hope to beat results of Gastal & Oliveira, Siggraph 2011).
- Extending to mesh processing (in a hope to beat results of Chuang & Kazhdan, Siggraph 2011).