Best apertures from entropy minimization

Bart M. ter Haar Romeny, PhD
Eindhoven University of Technology

Excerpt from the book "Front-End Vision & Multi-Scale Image Analysis", Kluwer Academic Publishers, 2004.

```
<< MathVisionTools`;
Off[Solve::ifun];
SetOptions[RasterPlot, Frame -> False];
SetOptions[Plot3D, PlotRange → All];
Off[GaussianDerivative::"scalefail"];
gD[im_, nx_, ny_, σ_] := GaussianDerivative[{σ^2/2, nx}, {σ^2/2, ny}][im];
SetOptions[RasterPlot, Frame -> False];
```


MathVisionTools` is a package written at Eindhoven University of Technology. It contains the functions GaussianDerivative and RasterPlot, used below. It can be downloaded from www.mathvisiontools.net.

Derivation

The derivation is based on:
M. Nielsen, "From paradigm to algorithms in computer vision". PhD thesis, Datalogisk Institut Kopenhagen University, Denmark, Dept. of Computer Science, Universitetsparken 1, DK-2100 Kopenhagen 0, Denmark, 1995. ISSN 0107-8283.

We do a measurement → finite aperture.
All locations treated similar → translation invariance.
Measurement is linear → superposition principle.

→ The observation must be a *convolution*: $h(x) = \int_{-\infty}^{\infty} L(\alpha)\, g(x-\alpha)\, d\alpha$.
$L(x)$ is the luminance, $g(x)$ is our aperture, $h(x)$ the result of our measurement.

A. The aperture function $g(x)$ should be a *normalized* filter: $\int_{-\infty}^{\infty} g(x)\, dx = 1$.

B. The *mean* of the filter $g(x)$ is $\int_{-\infty}^{\infty} x\, g(x)\, dx = x_0 = 0$.

C. The *width* is the variance: $\int_{-\infty}^{\infty} x^2\, g(x)\, dx = \sigma^2$.

The information or entropy of our filter is by definition: $H = \int_{-\infty}^{\infty} -g(x) \ln g(x)\, dx$.

We look for the $g(x)$ for which the entropy is minimal, *given the constraints*:
$\int_{-\infty}^{\infty} g(x)\, dx = 1$, $\int_{-\infty}^{\infty} x\, g(x)\, dx = 0$ and $\int_{-\infty}^{\infty} x^2\, g(x)\, dx = \sigma^2$.

When we want to find a minimum under a set of given constraints, we apply a standard mathematical technique named *variational methods*.

The energy *E* becomes:

$$E = \int_{-\infty}^{\infty} -g(x) \ln g(x)\, dx + \lambda_1 \int_{-\infty}^{\infty} g(x)\, dx + \lambda_2 \int_{-\infty}^{\infty} x\, g(x)\, dx + \lambda_3 \int_{-\infty}^{\infty} x^2\, g(x)\, dx$$

and is minimum when $\frac{\partial E}{\partial g} = 0$. The λ's are the *Lagrange multipliers*.

```
Clear[g];
<< VariationalMethods`;
var = VariationalD[-g[x] Log[g[x]] + λ1 g[x] + λ2 x g[x] + λ3 x^2 g[x], g[x], x]
```

$$-1 + \lambda 1 + x\, \lambda 2 + x^2\, \lambda 3 - \text{Log}[g[x]]$$

```
g[x_] = First[g[x] /. Solve[var == 0, g[x]]]
```

$$e^{-1+\lambda 1+x\, \lambda 2+x^2\, \lambda 3}$$

An important first finding is that *g* is an exponential function.

λ_3 must be negative, otherwise the function explodes, which is physically unrealistic.
The constraint equations are now:

```
eqn1 = Simplify[∫_-∞^∞ g[x] dx == 1, λ3 < 0]
```

$$e\, \sqrt{-\lambda 3} == e^{\lambda 1 - \frac{\lambda 2^2}{4\, \lambda 3}}\, \sqrt{\pi}$$

```
eqn2 = Simplify[∫_-∞^∞ x g[x] dx == 0, λ3 < 0]
```

$$e^{\lambda 1 - \frac{\lambda 2^2}{4\, \lambda 3}}\, \lambda 2 == 0$$

```
eqn3 = Simplify[∫_-∞^∞ x^2 g[x] dx == σ^2, λ3 < 0]
```

$$\frac{e^{-1+\lambda 1-\frac{\lambda 2^2}{4\, \lambda 3}}\, \sqrt{\pi}\ (\lambda 2^2 - 2\, \lambda 3)}{4\ (-\lambda 3)^{5/2}} == \sigma^2$$

Now we can solve for all three λ's:

```
solution = Solve[{eqn1, eqn2, eqn3}, {λ1, λ2, λ3}, Method → "Legacy"]
```

$$\left\{\left\{\lambda 1 \to \frac{1}{4}\, \text{Log}\left[\frac{e^4}{4\, \pi^2\, \sigma^4}\right],\ \lambda 2 \to 0,\ \lambda 3 \to -\frac{1}{2\, \sigma^2}\right\}\right\}$$

```
g[x_, σ_] = Simplify[g[x] /. Flatten[solution], σ > 0]
```

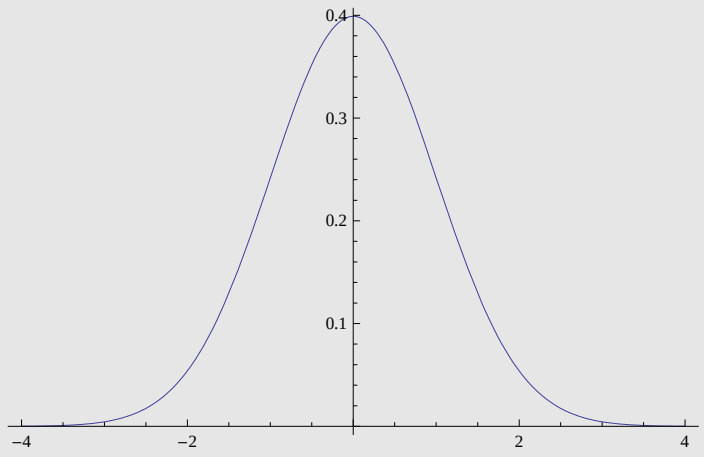
$$\frac{e^{-\frac{x^2}{2\, \sigma^2}}}{\sqrt{2\, \pi}\ \sigma}$$

which is the Gaussian function. A beautiful result. We have found the Gaussian as the *unique* solution to the set of constraints, which in principle are a formal statement of the *uncommittment* of the observation.

There are 11 known axiomatic derivations of the Gaussian kernel as the optimal aperture for uncommitted observations [Weickert 2002].

Relaxing the constraints gives other families, e.g.:
- separability: Poisson scale-space [Duits, Felsberg]
- tuned to spatial frequency: Gabor kernels

```
Plot[g[x, 1], {x, -4, 4}]
```

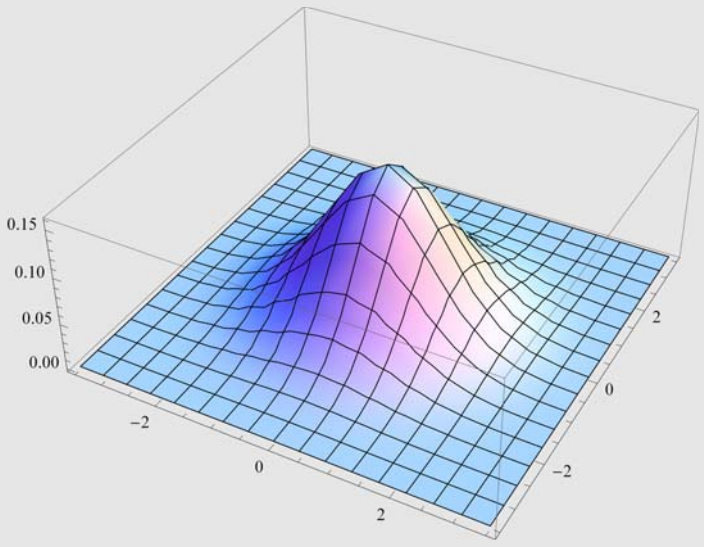


Derivatives of sampled, observed data

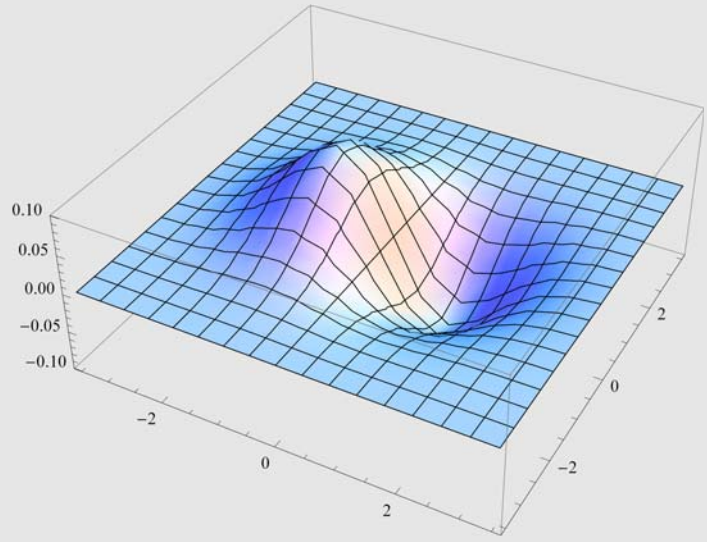
All *partial derivatives* of the Gaussian kernel are solutions too of the diffusion equation.

$$g := \frac{1}{2\,\pi\,\sigma^2}\,\text{Exp}\left[-\frac{x^2+y^2}{2\,\sigma^2}\right];\ \sigma = 1;$$

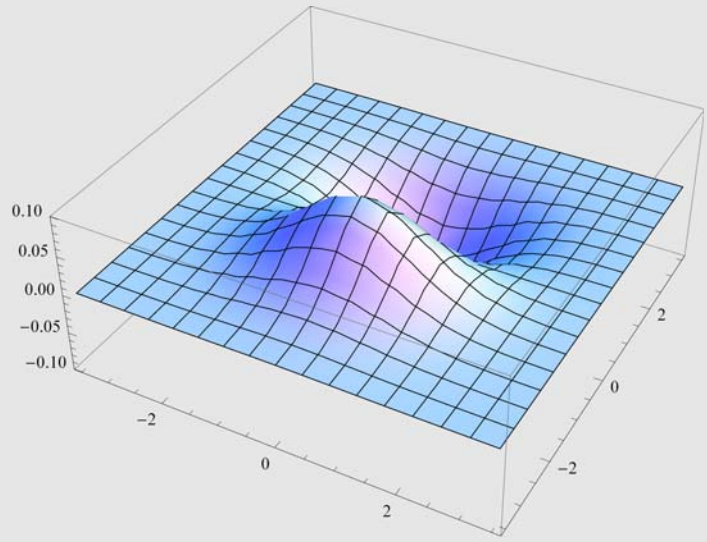
```
Plot3D[g, {x, -3.5, 3.5}, {y, -3.5, 3.5}]
```



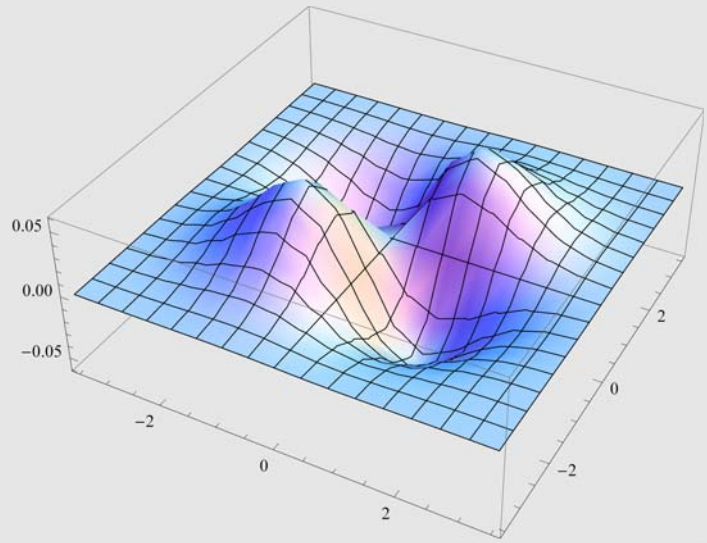
```
Plot3D[Evaluate[∂x g], {x, -3.5, 3.5}, {y, -3.5, 3.5}]
```



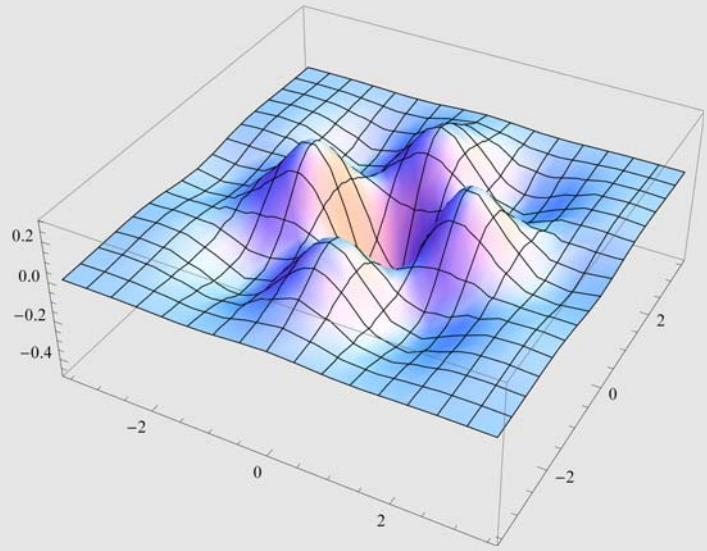

```
Plot3D[Evaluate[∂y g], {x, -3.5, 3.5}, {y, -3.5, 3.5}]
```




```
Plot3D[Evaluate[∂x (∂y g)], {x, -3.5, 3.5}, {y, -3.5, 3.5}]
```



```
Plot3D[Evaluate[∂x,x,x,x (∂y,y g)], {x, -3.5, 3.5}, {y, -3.5, 3.5}]
```



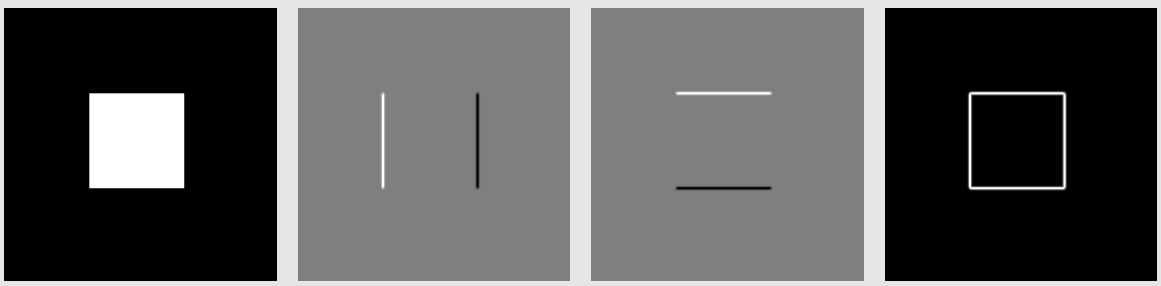
The derivative of the observed data is
$\frac{\partial}{\partial x}\{L_0(x,\ y)\otimes G(x,\ y;\sigma)\}$, which can be rewritten as $L_0(x,\ y)\otimes\frac{\partial}{\partial x}G(x,\ y;\sigma)$.

■ Differentiation of discrete data is convolution with a Gaussian derivative

- Differentiation and observation are done in a single step: convolution with a Gaussian derivative kernel.
- Differentiation is now done by *integration*, namely by the convolution integral.
- The Gaussian kernel is the *physical* analogue of a *mathematical* point, the Gaussian derivative kernels are the physical analogons of the mathematical differential operators. Equivalence is reached for the limit when the scale of the Gaussian goes to zero:

The function **gD[im,nx,ny,σ]** implements a convolution with a Gaussian derivative:

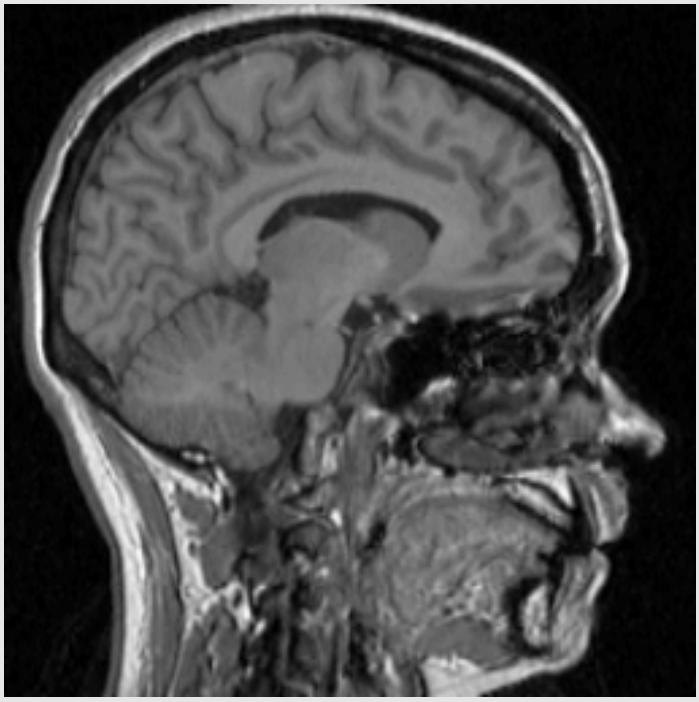
```
im = Table[If[80 < x < 170 && 80 < y < 170, 1, 0], {y, 1, 256}, {x, 1, 256}];
σ = 1;
imx = gD[im, 1, 0, σ]; imy = gD[im, 0, 1, σ];
grad = √(imx^2 + imy^2) ;
p1 = RasterPlot /@ {im, imx, imy, grad};
GraphicsGrid[{p1}, ImageSize → 600]
```



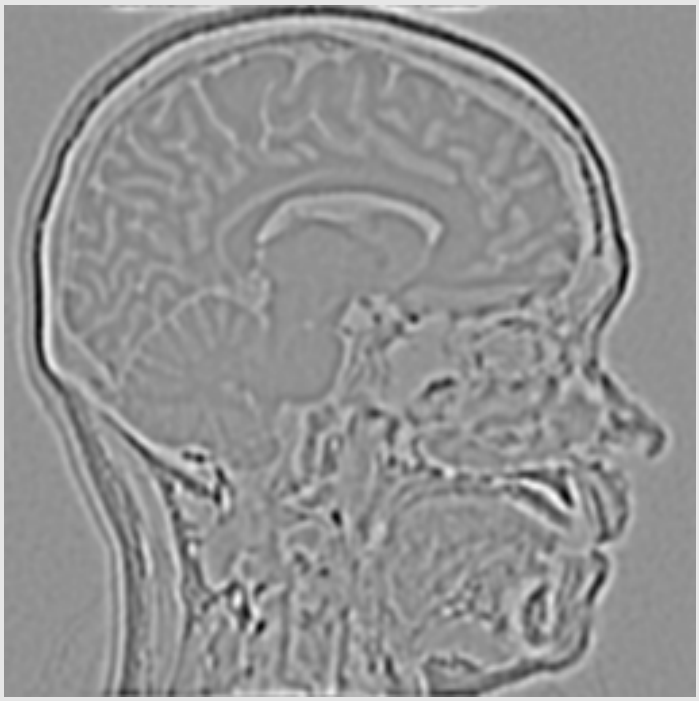
The first order derivative of an image gives edges. Left: original test image $L(x, y)$, resolution 256^2. Second: the derivative with respect to x: $\frac{\partial L}{\partial x}$ at scale $\sigma = 1$ pixel. Note the positive and negative edges. Third: the derivative with respect to y: $\frac{\partial L}{\partial y}$ at scale $\sigma = 1$ pixel. Right: the gradient $\sqrt{\left(\frac{\partial L}{\partial x}\right)^2+\left(\frac{\partial L}{\partial x}\right)^2}$ at a scale of $\sigma = 1$ pixel which gives all edges.

```
im = ImageData[ColorConvert[Import[
      "http://bmia.bmt.tue.nl/Education/Courses/FEV/course/testimages/mr256.gif"]
      , "Grayscale"], "Byte"];
```

```
p1 = RasterPlot[im]
```

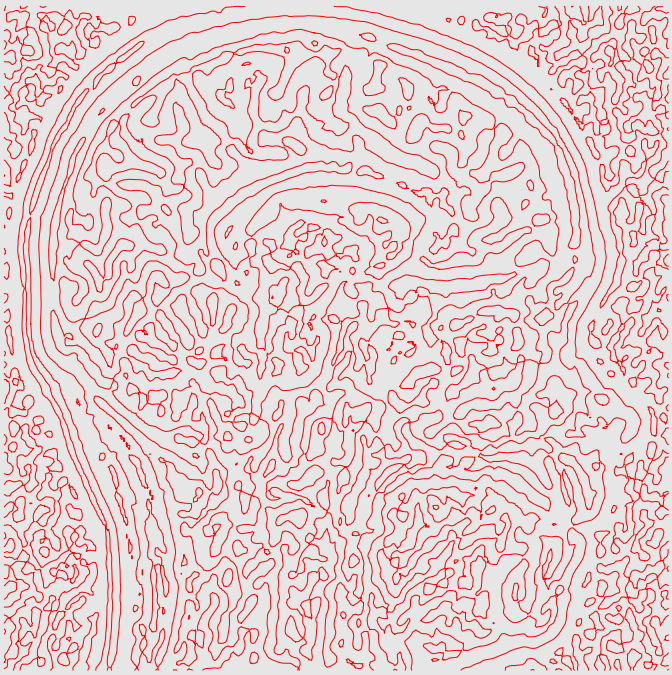


```
Lxx = im_{G1}^{(2,0)};
Lyy = im_{G1}^{(0,2)};
RasterPlot[Lxx + Lyy]
```

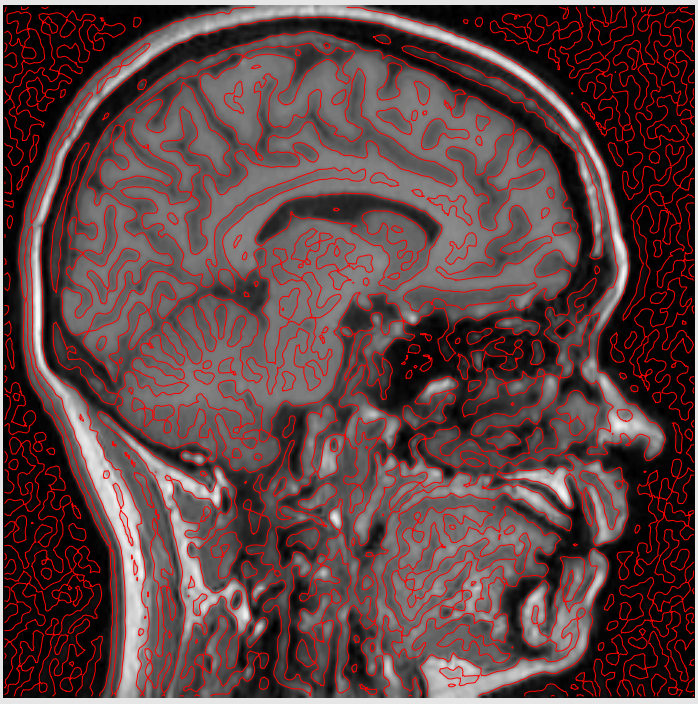


The Laplacian of the image, as seen by a center-surround cell of scale σ=1 pixel.

```
p2 = ListContourPlot[Reverse[Lxx + Lyy], Contours → {0},
  ContourStyle → Red, ContourShading → None, Frame → None]
```



```
Show[p1, p2]
```



- The Gaussian kernel is the unique kernel that generates no *spurious resolution*. It is the blown-up physical *point operator*, the Gaussian derivatives are the blown-up physical *multi-scale derivative operators*.

```
Import[
 "http://bmia.bmt.tue.nl/Education/Courses/FEV/book/images/blown-upddx.jpg",
 ImageSize → 300]
```

- Because convolving is an integration, the Gaussian kernel has by definition a strong *regularizing* effect.
- Recently some interesting papers have shown the complete equivalence of Gaussian scale space regularization with a number of other methods for regularization such as splines, thin plate splines, graduated convexity etc. [Scherzer2000a, Nielsen1996b, Nielsen1997b].
- The set of Gaussian derivative kernels (including the zeroth order derivative: the Gaussian kernel itself) forms a *complete* set of derivatives. This set is sometimes referred to as the *N-jet*.

  A scale-space is a stack of 2D images, where scale is the third dimension.

  And here are
  - a scale-space of the intensity (no derivatives, only blurred);
  - a scale-space of the Laplacian ($\frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2}$).

```
im = ImageData[ColorConvert[Import[
      "http://bmia.bmt.tue.nl/Education/Courses/FEV/book/images/mr256.gif"],
     "Grayscale"], "Byte"];
```

```
GraphicsGrid[{Table[RasterPlot[gD[im, 0, 0, Eτ]], {τ, 0, 2.1, .3}], Table[
    RasterPlot[gD[im, 2, 0, Eτ] + gD[im, 0, 2, Eτ]], {τ, 0, 2.1, .3}]}, ImageSize → 600]
```