Image Processing and Computer Vision (IPCV)

Prof. J. Weickert
Mathematical Image Analysis Group

Summer term 2019
Saarland University

Example Solutions for Homework Assignment 9 (H9)

Problem 1 (Continuous Nonquadratic Variational Methods)

(a) For an energy functional of the form

$$E(u) = \int_b^a F(u, u_x)$$

the associated Euler-Lagrange equation is given by

$$F_u - \frac{\partial}{\partial x} F_{u_x} = 0 \; .$$

In our case, the corresponding derivatives read

$$\begin{aligned} F_u &= u - f \;, \\ F_{u_x} &= \alpha\, \lambda^2 \frac{1}{2\,\sqrt{1 + u_x^2/\lambda^2}} \frac{2\,u_x}{\lambda^2} = \alpha \left(\frac{1}{\sqrt{1 + u_x^2/\lambda^2}}\, u_x \right) \; . \end{aligned}$$

Taking the derivative with respect to x of the second term yields

$$\frac{\partial}{\partial_x} F_{u_x} = \alpha \frac{\partial}{\partial_x} \left(\frac{1}{\sqrt{1 + u_x^2/\lambda^2}}\, u_x \right) \; .$$

If desired this can be further simplified to

$$\begin{aligned} \alpha \frac{\partial}{\partial_x} \left(\frac{1}{\sqrt{1 + u_x^2/\lambda^2}}\, u_x \right) &= \alpha \left(\frac{u_{xx} \sqrt{1 + u_x^2/\lambda^2} - u_x \frac{1}{2\sqrt{1+u_x^2/\lambda^2}} \frac{2\,u_x}{\lambda^2} u_{xx}}{1 + u_x^2/\lambda^2} \right) \\ &= \alpha \left(\frac{\frac{u_{xx}}{\sqrt{1+u_x^2/\lambda^2}} (1 + u_x^2/\lambda^2) - \frac{u_{xx}}{\sqrt{1+u_x^2/\lambda^2}} \frac{(u_x)^2}{\lambda^2}}{1 + u_x^2/\lambda^2} \right) \\ &= \alpha \frac{u_{xx}}{(1 + u_x^2/\lambda^2)^{\frac{3}{2}}} \; . \end{aligned}$$

Putting everything together, we obtain the Euler-Lagrange equation

$$u - f - \alpha \frac{\partial}{\partial_x} \left(\frac{1}{\sqrt{1 + u_x^2/\lambda^2}} \, u_x \right) = 0 \, .$$

(b) The boundary conditions are given as

$$\begin{aligned} & F_{u_x} = 0 \\ \Leftrightarrow \quad & \alpha \left(\frac{1}{\sqrt{1 + u_x^2/\lambda^2}} \, u_x \right) = 0 \\ \Leftrightarrow \quad & u_x = 0 \end{aligned}$$

The last step is possible, since $\sqrt{1 + u_x^2/\lambda^2} > 0$. Thus, we have reflecting conditions on the boundaries a and b.

(c) It is sufficient to prove that $E(u)$ is strictly convex, as it follows that $E(u)$ has a single global minimum which is the unique solution of the Euler-Lagrange equation. (We can assume that E is coercive).

First we show the convexity of the data term of our functional, which is given by

$$\frac{1}{2} (u - f)^2 =: D(u) \, .$$

We exploit the fact that a function is strictly convex if its second derivative is positive. Thus we compute

$$\frac{\partial^2}{\partial u} \left(\frac{1}{2} (u - f)^2 \right) = \frac{\partial}{\partial u} (u - f) = 1 > 0$$

which shows the convexity. Now we consider the smoothness term, given by

$$\alpha \, \lambda^2 \sqrt{1 + u_x^2/\lambda^2} =: S(u_x) \, .$$

In part (a) we already computed the first derivative

$$\frac{\partial}{\partial u_x} S(u_x) = \frac{\partial}{\partial u_x} \left(\alpha \, \lambda^2 \sqrt{1 + u_x^2/\lambda^2} \right) = \frac{\alpha \, u_x}{\sqrt{1 + u_x^2/\lambda^2}} \, .$$

The second derivative is given by

$$\begin{aligned} \frac{\partial^2}{\partial u_x^2} S(u_x) &= \frac{\partial}{\partial u_x} \left(\frac{\alpha \, u_x}{\sqrt{1 + u_x^2/\lambda^2}} \right) = \alpha \left(\frac{\sqrt{1 + u_x^2/\lambda^2} - u_x \frac{u_x}{\lambda^2 \sqrt{1 + u_x^2/\lambda^2}}}{1 + u_x^2/\lambda^2} \right) \\ &= \alpha \, \frac{1}{(1 + u_x^2/\lambda^2)^{3/2}} \end{aligned}$$

As $\alpha > 0$ and $1 + u_x^2/\lambda^2 \geq 1$, the second derivative is positive, thus also our smoothness term is strictly convex.

We now use these results to show the strict convexity of our energy functional. A functional $E : X \to Y$ is strictly convex if it holds $\forall u, v \in X\ \forall \beta \in]0,1[$:

$$E(\beta u + (1-\beta)v) < \beta E(u) + (1-\beta)E(v)$$

In our case, we have

$$\begin{aligned}
&E\Big(\beta\, u + (1-\beta)\, v\Big) \\
&= \int_a^b D\Big(\beta\, u + (1-\beta)\, v\Big) + S\Big(\big(\beta u + (1-\beta) v\big)_x\Big)\, dx \\
&= \int_a^b D\Big(\beta\, u + (1-\beta)\, v\Big)\, dx + \int_a^b S\Big(\big(\beta u + (1-\beta) v\big)_x\Big)\, dx \\
&= \int_a^b D\Big(\beta\, u + (1-\beta)\, v\Big)\, dx + \int_a^b S\Big(\beta\, u_x + (1-\beta)\, v_x\Big)\, dx
\end{aligned} \tag{1}$$

where in the last step we have used the linearity of the differential operator. Due to the convexity of D we have

$$D\Big(\beta\, u + (1-\beta)\, v\Big) < \beta\, D(u) + (1-\beta)\, D(v)$$

As $D(w)$ is non-negative $\forall w$ and $a \leq b$, it follows that

$$\int_a^b D\Big(\beta\, u + (1-\beta)\, v\Big)\, dx < \int_a^b \beta\, D(u) + (1-\beta)\, D(v)\, dx \tag{2}$$

Analogously we want to derive an inequality for the second term. As S is strictly convex, it holds that

$$S\Big(\beta\, u_x + (1-\beta)\, v_x\Big) < \beta S(u_x) + (1-\beta) S(v_x)$$

Again we use the fact that $S(w)$ is non-negative $\forall w$, which leads us to

$$\int_a^b S\Big(\beta\, u_x + (1-\beta)\, v_x\Big)\, dx < \int_a^b \beta\, S(u_x) + (1-\beta)\, S(v_x)\, dx \tag{3}$$

Finally, we combine equation (1) with inequalities (2) and (3)

$$
\begin{aligned}
& E\Big(\beta\, u + (1-\beta)\, v\Big) \\
&= \int_a^b D\Big(\beta\, u + (1-\beta)\, v\Big) + S\Big(\big(\beta u + (1-\beta) v\big)_x\Big)\, dx \\
&< \int_a^b \beta\, D(u) + (1-\beta)\, D(v)\, dx + \int_a^b \beta S(u_x) + (1-\beta)\, S(v_x)\, dx \\
&= \beta \int_a^b D(u) + S(u_x)\, dx + (1-\beta) \int_a^b D(v) + S(v_x)\, dx \\
&= \beta\, E(u) + (1-\beta)\, E(v)
\end{aligned}
$$

which concludes the proof.

Problem 2 (Discrete Variational Methods)

(a) In analogy to the functional considered in Problem 1, we write down a discrete version of $E(u)$ as follows:

$$E(u) := \frac{1}{2}\sum_{k=1}^{N}(u_k - f_k)^2 + \alpha \sum_{k=1}^{N-1} \lambda^2 \sqrt{1 + \frac{(u_{k+1}-u_k)^2}{\lambda^2 h^2}}\,.$$

Here, we assume that the finite forward difference and the length of the signal f, i.e., $N = \frac{b-a}{h}$, depend on the pixel distance $h > 0$, which is often set to 1 in practice.

(b) The minimiser of the discrete functional $E(u)$ necessarily satisfies the nonlinear system of equations $\dfrac{\partial E(u)}{\partial u_k} = 0$ for all $k = 1, \ldots, N$. Thus, we have to calculate partial derivatives distinguishing boundary pixels from inner pixels:

$$\frac{\partial E(u)}{\partial u_1} = u_1 - f_1 - \frac{\alpha}{h^2}\frac{u_2 - u_1}{\sqrt{1 + \dfrac{(u_2-u_1)^2}{\lambda^2 h^2}}}\,, \qquad \text{(for } k = 1\text{)},$$

$$\frac{\partial E(u)}{\partial u_k} = u_k - f_k - \frac{\alpha}{h^2}\frac{u_{k+1} - u_k}{\sqrt{1 + \dfrac{(u_{k+1}-u_k)^2}{\lambda^2 h^2}}} + \frac{\alpha}{h^2}\frac{u_k - u_{k-1}}{\sqrt{1 + \dfrac{(u_k-u_{k-1})^2}{\lambda^2 h^2}}}\,, \qquad \text{(for } k = 2, \ldots, N-1\text{)},$$

$$\frac{\partial E(u)}{\partial u_N} = u_N - f_N + \frac{\alpha}{h^2}\frac{u_N - u_{N-1}}{\sqrt{1 + \dfrac{(u_N-u_{N-1})^2}{\lambda^2 h^2}}}\,, \qquad \text{(for } k = N\text{)}.$$

Problem 3 (Fourier Analysis of Linear Filters)

(a) For each filter the Fourier transform of the signal u is represented by a multiple of the Fourier transform of f, i.e. $\hat{u} = g \cdot \hat{f}$ with filter specific functions f and g.

(i) 1-D discrete regularisation with grid size h:

$$-\frac{\alpha}{h^2}u(x-h) + \left(1 + 2\frac{\alpha}{h^2}\right)u(x) - \frac{\alpha}{h^2}u(x+h) = f(x)$$

Compute the Fourier transform of f:

$$\begin{aligned}
\hat{f}(y) &= \int_{-\infty}^{\infty} \left(-\frac{\alpha}{h^2}u(x-h) + \left(1 + 2\frac{\alpha}{h^2}\right)u(x) - \frac{\alpha}{h^2}u(x+h)\right) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \cdot \exp(-i2\pi xy)dx \\
&= -\frac{\alpha}{h^2}\int_{-\infty}^{\infty} u(x-h)\exp(-i2\pi xy)dx \\
&\quad + \left(1 + 2\frac{\alpha}{h^2}\right)\int_{-\infty}^{\infty} u(x)\exp(-i2\pi xy)dx \\
&\quad - \frac{\alpha}{h^2}\int_{-\infty}^{\infty} u(x+h)\exp(-i2\pi xy)dx \\
&= -\frac{\alpha}{h^2}\hat{u}(y-h) + \left(1 + 2\frac{\alpha}{h^2}\right)\hat{u}(y) - \frac{\alpha}{h^2}\hat{u}(y+h) \\
&\overset{(1)}{=} -\frac{\alpha}{h^2}\exp(-i2\pi yh)\hat{u}(y) + \left(1 + 2\frac{\alpha}{h^2}\right)\hat{u}(y) - \frac{\alpha}{h^2}\exp(i2\pi yh)\hat{u}(y) \\
&= \left(1 + 2\frac{\alpha}{h^2} - \frac{\alpha}{h^2}(\exp(-i2\pi yh) + \exp(i2\pi yh)\right)\hat{u}(y) \\
&\overset{(2)}{=} \left(1 + 2\frac{\alpha}{h^2}(1 - \cos(2\pi yh))\right)\hat{u}(y)
\end{aligned}$$

In step (1) we use the shift theorem and for (2) holds by Euler's formula:

$$\exp(-ix) + \exp(ix) = \cos(x) - i\sin(x) + \cos(x) + i\sin(x) = 2\cos(x) \ .$$

Rearranging the resulting equation gives us

$$\hat{u}(y) = \left(1 + 2\frac{\alpha}{h^2}(1 - \cos(2\pi yh))\right)^{-1}\hat{f}(y)$$

and thus the final result is the function g with

$$g(y) = \left(1 + 2\frac{\alpha}{h^2}(1 - \cos(2\pi yh))\right)^{-1} \ .$$

(ii) 1-D continuous regularisation is defined by the energy functional

$$E_f(u) = \frac{1}{2}\int_{\Omega} \underbrace{(u-f)^2 + \alpha(u')^2}_{=F(x,u,u')}\, dx, \qquad \Omega \subset \mathbb{R}\,.$$

The Euler-Lagrange equation for F yields the equation from the exercise sheet:

$$\begin{aligned} 0 &= u(x) - f(x) - \alpha u''(x) \\ \Leftrightarrow f(x) &= u(x) - \alpha u''(x)\,. \end{aligned}$$

Now we can compute the Fourier transform of f exploiting the linearity of the Fourier transform (1):

$$\begin{aligned} \hat{f}(y) &\overset{(1)}{=} \mathcal{F}[u - \alpha u''](y) = \mathcal{F}[u(x)](y) - \alpha\mathcal{F}[u''](y) \\ &\overset{(2)}{=} \hat{u}(y) - \alpha(i2\pi y)^2\hat{u}(y) = (1 + \alpha 4\pi^2 y^2)\hat{u}(y)\,. \end{aligned}$$

Equality (2) holds because of the differentiation rule:

$$\mathcal{F}\left[\frac{\partial^n}{\partial x^n} f\right](y) = (i2\pi y)^n \mathcal{F}[f](y)$$

Thus the final result for g is

$$g(y) = (1 + \alpha 4\pi^2 y^2)^{-1}\,.$$

(iii) Convolution with a 1-D Gaussian kernel is given by:

$$u = K_\sigma * f \quad \text{ with } K_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)\,.$$

Recall that convolution in the spatial domain is equivalent to multiplication in the Fourier domain (*convolution theorem*) and that the Fourier transform of the Gaussian $K_\sigma(x)$ was given in Lecture 4, Slide 29. Thus, we have:

$$\hat{u}(y) = \hat{f}(y) \cdot \mathcal{F}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)\right](y) \overset{\text{L4 S28}}{=} \exp\left(-\frac{(2\pi y)^2}{2\sigma^{-2}}\right) \cdot \hat{f}(y)\,.$$

(iv) Consider the 1-D explicit scheme for nonlinear diffusion which was also a topic of H8:

$$u_i^{k+1} = u_i^k + \frac{\tau}{h}\left(g_{i+\frac{1}{2}}^k \frac{u_{i+1}^k - u_i^k}{h} - g_{i-\frac{1}{2}}^k \frac{u_i^k - u_{i-1}^k}{h}\right)$$

Applying $g = 1$ and the initial condition $u^0 = f$ to this scheme gives the filter result after one iteration step with constant diffusivity:

$$u(x) = \frac{\tau}{h^2} f(x-h) + \left(1 - 2\frac{\tau}{h^2}\right) f(x) + \frac{\tau}{h^2} f(x+h)$$

As this equation is similar to (i) we can compute the Fourier transform of u analogously:

$$\begin{aligned}
\hat{u}(y) &= \int_{-\infty}^{\infty} \left(\frac{\tau}{h^2} f(x-h) + \left(1 - 2\frac{\tau}{h^2}\right) f(x) + \frac{\tau}{h^2} f(x+h)\right) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \cdot \exp(-i2\pi xy) dx \\
&= \frac{\tau}{h^2} \hat{f}(y-h) + \left(1 - 2\frac{\tau}{h^2}\right) \hat{f}(y) + \frac{\tau}{h^2} \hat{f}(y-h) \\
&\overset{(1)}{=} \frac{\tau}{h^2} \exp(-i2\pi yh) \hat{f}(y) + \left(1 - 2\frac{\tau}{h^2}\right) \hat{f}(y) + \frac{\tau}{h^2} \exp(i2\pi yh) \hat{f}(y) \\
&\overset{(2)}{=} \left(1 + 2\frac{\tau}{h^2}(\cos(2\pi yh) - 1)\right) \hat{f}(y)
\end{aligned}$$

with (1),(2) as in (i).

(b) We want to prove that for $\alpha = \tau = \frac{1}{2}\sigma^2$ and $h \to 0$ the quadratic Taylor polynomial in 0 is equal for the corresponding functions g from (i)-(iv), which means, that the four different filters give approximatively the same results. First, consider the Taylor expansion of g in $a \in \mathbb{R}$:

$$g(x) = g(a) + \frac{g'(a)}{1!}(x-a) + \frac{g''(a)}{2!}(x-a)^2 + \cdots \;.$$

As we are only interested in the quadratic Taylor polynomial with $a = 0$, for each g we have to compute

$$g(x) \approx g(0) + g'(0)x + \frac{g''(0)}{2} x^2 \;.$$

(i) 1-D discrete regularisation with grid size h:

$$\begin{aligned}
g(y) &= \left(1 + 2\frac{\alpha}{h^2}(1 - \cos(2\pi yh))\right)^{-1} \\
g'(y) &= \frac{4\pi\alpha \sin(2\pi yh)}{h\left(1 + 2\frac{\alpha}{h^2}(1 - \cos(2\pi yh))\right)^2} \\
g''(y) &= \frac{-8\pi^2\alpha \cos(2\pi yh)}{\left(1 + 2\frac{\alpha}{h^2}(1 - \cos(2\pi yh))\right)^2} + \frac{32\pi^2\alpha^2 \sin(2\pi yh)^2}{h^2\left(1 + 2\frac{\alpha}{h^2}(1 - \cos(2\pi yh))\right)^3}
\end{aligned}$$

and thus

$$
\begin{aligned}
& g(y) + g'(y)x + \frac{g''(y)}{2}x^2 \\
= \quad & \frac{1}{1+2\frac{\alpha}{h^2}(1-\cos(2\pi yh))} + \frac{4\pi\alpha\sin(2\pi yh)}{h\left(1+2\frac{\alpha}{h^2}(1-\cos(2\pi yh))\right)^2}x \\
& + \frac{1}{2}\left(\frac{-8\pi^2\alpha\cos(2\pi yh)}{\left(1+2\frac{\alpha}{h^2}(1-\cos(2\pi yh))\right)^2} + \frac{32\pi^2\alpha^2\sin(2\pi yh)^2}{h^2\left(1+2\frac{\alpha}{h^2}(1-\cos(2\pi yh))\right)^3}\right)x^2 \\
\overset{y=0}{=} \quad & 1 - 4\pi^2\alpha x^2 \overset{\alpha=\frac{\sigma^2}{2}}{=} 1 - 2\sigma^2\pi^2 x^2 \ .
\end{aligned}
$$

(ii) 1-D continuous regularisation:

$$
\begin{aligned}
g(y) &= (1+\alpha 4\pi^2 y^2)^{-1} \\
g'(y) &= \frac{-8\pi^2\alpha y}{(1+\alpha 4\pi^2 y^2)^2} \\
g''(y) &= \frac{128\pi^4\alpha^2 y^2}{(1+\alpha 4\pi^2 y^2)^3} - \frac{8\alpha\pi^2}{(1+\alpha 4\pi^2 y^2)^2} = \frac{8\alpha\pi^2(12\pi^2\alpha y^2 - 1)}{(1+\alpha 4\pi^2 y^2)^3}
\end{aligned}
$$

and thus

$$
\begin{aligned}
& g(y) + g'(y)x + \frac{g''(y)}{2}x^2 \\
&= \frac{1}{1+\alpha 4\pi^2 y^2} + \frac{-8\pi^2\alpha y}{(1+\alpha 4\pi^2 y^2)^2}x + \frac{8\alpha\pi^2(12\pi^2\alpha y^2 - 1)}{2(1+\alpha 4\pi^2 y^2)^3}x^2 \\
&\overset{y=0}{=} 1 - 4\alpha\pi^2 x^2 \overset{\alpha=\frac{\sigma^2}{2}}{=} 1 - 2\sigma^2\pi^2 x^2 \ .
\end{aligned}
$$

(iii) Convolution with 1-D Gaussian kernel K_σ:

$$
\begin{aligned}
g(y) &= \exp\left(-\frac{(2\pi y)^2}{2\sigma^{-2}}\right) = \exp\left(-2\pi^2\sigma^2 y^2\right) \\
g'(y) &= -4\pi^2\sigma^2 y \exp\left(-2\pi^2\sigma^2 y^2\right) \\
g''(y) &= 16\pi^4\sigma^4 y^2 \exp\left(-2\pi^2\sigma^2 y^2\right) - 4\pi^2\sigma^2 \exp\left(-2\pi^2\sigma^2 y^2\right)
\end{aligned}
$$

and thus

$$
\begin{aligned}
& g(y) + g'(y)x + \frac{g''(y)}{2}x^2 \\
&= (1 - 4\pi^2\sigma^2 yx + \frac{16\pi^4\sigma^4 y^2 - 4\pi^2\sigma^2}{2}x^2)\exp\left(-2\pi^2\sigma^2 y^2\right) \\
&\overset{y=0}{=} 1 - 2\pi^2\sigma^2 x^2 \ .
\end{aligned}
$$

(iv) 1-D explicit scheme for linear diffusion after one iteration step:

$$\begin{aligned} g(y) &= 1 + 2\frac{\tau}{h^2}(1 - \cos(2\pi y h)) \\ g'(y) &= -\frac{4\pi\tau}{h}\sin(2\pi h y) \\ g''(y) &= -8\pi^2\tau\cos(2\pi h y) \end{aligned}$$

and thus

$$\begin{aligned} & g(y) + g'(y)x + \frac{g''(y)}{2}x^2 \\ &= 1 + 2\frac{\tau}{h^2}(1 - \cos(2\pi y h)) - \frac{4\pi\tau}{h}\sin(2\pi h y)x - 4\pi^2\tau\cos(2\pi h y)x^2 \\ &\overset{y=0}{=} 1 - 4\pi^2\tau x^2 \overset{\tau=\frac{\sigma^2}{2}}{=} 1 - 2\sigma^2\pi^2 x^2\ . \end{aligned}$$

Problem 4 (Whittaker-Tikhonov Regularisation and Unsharp Masking)

(a) We supplement the code in the function `regularise` for the numerator and denominator as given in lecture 17, slide 20.

```
/* copy initial image into working copy of f */
for (j=1; j<=ny; j++)
 for (i=1; i<=nx; i++)
     tmp[i][j] = f[i][j];


/* ---- iterate until stopping criterion is satisfied ---- */

k = 0;
do
{
    /* write last version of working copy tmp into u */
    for (j=1; j<=ny; j++)
     for (i=1; i<=nx; i++)
         u[i][j] = tmp[i][j];

    /* mirror boundaries */
    dummies(u, nx, ny);

    /* compute result for the iteration step k+1 (store it in tmp)
       and compute the residue w.r.t. the iteration step k */
    residue_k = 0.0;
```

```
    for (j=1; j<=ny; j++)
     for (i=1; i<=nx; i++)
         {
         numerator =  f[i][j] + alpha * ( ((i==1)  ? 0.0f : u[i-1][j])
                                         +((i==nx) ? 0.0f : u[i+1][j])
                                         +((j==1)  ? 0.0f : u[i][j-1])
                                         +((j==ny) ? 0.0f : u[i][j+1]) );

         denominator =  1.0f + alpha * ( ((i==1)  ? 0.0f : 1.0f)
                                        +((i==nx) ? 0.0f : 1.0f)
                                        +((j==1)  ? 0.0f : 1.0f)
                                        +((j==ny) ? 0.0f : 1.0f) );

         /* time saver */
         float res = numerator - denominator * u[i][j];

         residue_k += res * res;

         /* result for the iteration step k+1 */
         tmp[i][j] = numerator / denominator;
         }

    residue_k = sqrt (residue_k);
    if (k==0)
       residue_0 = residue_k;
    k++;
}
while (residue_k >= EPSILON * residue_0);
```

(b) The function `unsharp_masking` is completed by blurring the image f using the function `regularise` from task (a) and supplementing the unsharp masking formula as given in the problem task.

```
void unsharp_masking

     (float   **f,         /* image */
      long    nx,          /* pixel number in x direction */
      long    ny,          /* pixel number in x direction */
      float   alpha)       /* regulariser */

/*
 applies unsharp masking to u
*/

{
long   i, j;                 /* loop variables */
float  **u;                  /* blurred image */

printf("Applying unsharp masking...\n\n");
```

```
alloc_matrix (&u, nx+2, ny+2);

/* blur image by Whittaker-Tikhonov regularisation */
regularise(f,u,nx,ny,alpha);

/* unsharp masking */
for (j=1; j<=ny; j++)
  for (i=1; i<=nx; i++)
    f[i][j] = f[i][j] - (u[i][j] - f[i][j]);


disalloc_matrix (u, nx+2, ny+2);

return;

} /* unsharp_masking */
```

(c) *Left:* Blurred image `girl-blurred.pgm`.
Middle: Result after deblurring with unsharp masking (factor=5) and Whittaker-Tikhonov regularisation ($\alpha = 1.0$).
Right: Original image `girl.pgm`.

To better understand why this method works we can consider the following. We have the given signal $\boldsymbol{f}$, which is a blurred version of some original signal $\boldsymbol{g}$. Their difference $\boldsymbol{g}-\boldsymbol{f}$ are the details we have lost by blurring $\boldsymbol{g}$. What we want to do is to add them back to $\boldsymbol{f}$. This would result in the perfect recovery of $\boldsymbol{g}$ via

$$\boldsymbol{f}-(\boldsymbol{f}-\boldsymbol{g})=\boldsymbol{g}.$$

Since we do not know the original signal $\boldsymbol{g}$ or the difference $\boldsymbol{g}-\boldsymbol{f}$, we make an approximation. We consider $\boldsymbol{u}$, a blurred version of $\boldsymbol{f}$ and say that the details we removed, $\boldsymbol{f}-\boldsymbol{u}$, are approximately the original removed details $\boldsymbol{g}-\boldsymbol{f}$. We then add those details back to $\boldsymbol{f}$ and get the approximation

$$\boldsymbol{g}\approx\boldsymbol{f}-(\boldsymbol{u}-\boldsymbol{f}).$$

In Figure 1 we can see how this works when g is a 1-D signal, consisting of a single edge.

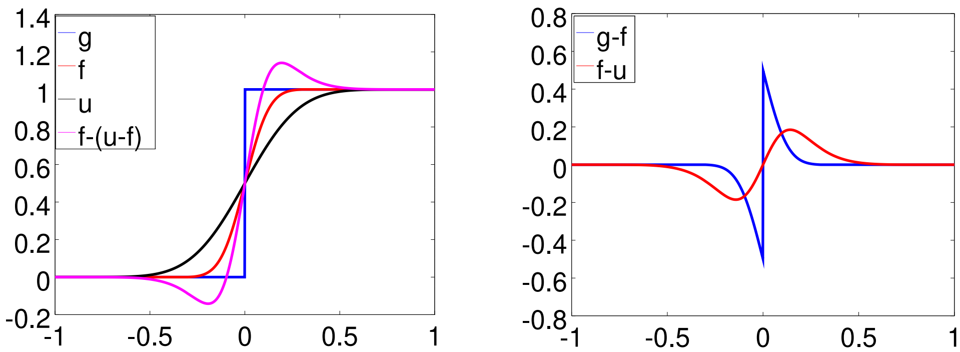

Figure 1: Left: The original signal $\boldsymbol{g}$ with a sharp edge, the blurred version $\boldsymbol{f}$, $\boldsymbol{u}$: the blurred version of $\boldsymbol{f}$, and the sharpened version. Right: The difference $\boldsymbol{g}-\boldsymbol{f}$ that we wanted to add back to $\boldsymbol{f}$, and our approximation $\boldsymbol{f}-\boldsymbol{u}$, that we did add back to $\boldsymbol{f}$. We see that the filtered signal has a sharper edge that $\boldsymbol{f}$, but it is not a perfect reconstruction of $\boldsymbol{g}$.