Prof. Dr.-Ing. A. Bruhn
Institute for Visualization and Interactive Systems
Department Intelligent Systems
University of Stuttgart

Example Solution for Homework Assignment 3

Problem 3.1 (Variational Methods) 10 Points

(a) The linearised assumption reads

$$f_{yx}u + f_{yy}v + f_{yt} = 0\ .$$

(b) A Horn-and-Schunck-like energy functional is given by

$$E(u,v) = \int_\Omega (f_{yx}u + f_{yy}v + f_{yt})^2 + \alpha\left(|\nabla u|^2 + |\nabla v|^2\right)\, dxdy$$

(c) The Euler-Lagrange equations can be computed directly:

$$\begin{aligned} 0 &= 2\cdot(f_{yx}u + f_{yy}v + f_{yt})\cdot f_{yx} - \alpha\cdot 2\cdot u_{xx} - \alpha\cdot 2\cdot u_{yy} \\ 0 &= 2\cdot(f_{yx}u + f_{yy}v + f_{yt})\cdot f_{yy} - \alpha\cdot 2\cdot v_{xx} - \alpha\cdot 2\cdot v_{yy} \end{aligned}$$

or, simplified,

$$\begin{aligned} 0 &= (f_{yx}u + f_{yy}v + f_{yt})\cdot f_{yx} - \alpha \text{ div } \nabla u \\ 0 &= (f_{yx}u + f_{yy}v + f_{yt})\cdot f_{yy} - \alpha \text{ div } \nabla v \end{aligned}$$

(d) A possible discretisation of the Euler-Lagrange equations using pixel coordinates $i = 1,\ldots,N$ and $j = 1,\ldots,M$ is given by

$$\begin{aligned} 0 = \;& \left(\frac{f_{i+1,j+1} - f_{i+1,j-1} - f_{i-1,j+1} + f_{i-1,j-1}}{4h_xh_y}\cdot u_{i,j} + \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{h_y^2}\cdot v_{i,j}\right. \\ & \left. + \frac{f^2_{i,j+1} - f^2_{i,j} - f^1_{i,j+1} + f^1_{i,j}}{h_yh_t}\right)\cdot\frac{f_{i+1,j+1} - f_{i+1,j-1} - f_{i-1,j+1} + f_{i-1,j-1}}{4h_xh_y} \\ & -\alpha\left(\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i-1,j}}{h_y^2}\right) \\ 0 = \;& \left(\frac{f_{i+1,j+1} - f_{i+1,j-1} - f_{i-1,j+1} + f_{i-1,j-1}}{4h_xh_y}\cdot u_{i,j} + \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{h_y^2}\cdot v_{i,j}\right. \\ & \left. + \frac{f^2_{i,j+1} - f^2_{i,j} - f^1_{i,j+1} + f^1_{i,j}}{h_yh_t}\right)\cdot\frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{h_y^2} \\ & -\alpha\left(\frac{v_{i+1,j} - 2v_{i,j} + v_{i-1,j}}{h_x^2} + \frac{v_{i,j+1} - 2v_{i,j} + v_{i-1,j}}{h_y^2}\right) \end{aligned}$$

Please note that we have not yet considered boundary conditions in this discretisation.

Alternative solution: We follow Slides 12 and 13 of Lecture 9, and make use of expressions $f_{yx,i}$, $f_{yy,i}$ and $f_{yz,i}$ that denote derivative approximations as the result of finite differences. For a set of pixels $i = 1, \ldots, N$ pixels, such a discretisation reads

$$\begin{aligned} 0 &= f_{yx,i}(f_{yx,i}u_i + f_{yy,i}v_i + f_{yz,i}) - \alpha \sum_{j \in \mathbb{N}(i)} (u_j - u_i) \ , \\ 0 &= f_{yy,i}(f_{yx,i}u_i + f_{yy,i}v_i + f_{yz,i}) - \alpha \sum_{j \in \mathbb{N}(i)} (v_j - v_i) \ , \end{aligned}$$

(e) Before setting up the iterative scheme, let us introduce some abbreviations:

$$\begin{aligned} a_{i,j} &= [f_{yx}]_{i,j} &&= \frac{f_{i+1,j+1} - f_{i+1,j-1} - f_{i-1,j+1} + f_{i-1,j-1}}{4h_x h_y} \\ b_{i,j} &= [f_{yy}]_{i,j} &&= \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{h_y^2} \\ c_{i,j} &= [f_{yt}]_{i,j} &&= \frac{f^2_{i,j+1} - f^2_{i,j} - f^1_{i,j+1} + f^1_{i,j}}{h_y h_t} \end{aligned}$$

This allows to rewrite the discrete Euler-Lagrange equations as follows:

$$\begin{aligned} 0 &= (a_{i,j}u_{i,j} + b_{i,j}v_{i,j} + c_{i,j})\, a_{i,j} - \alpha \sum_{l \in \{x,y\}} \sum_{\tilde{i},\tilde{j} \in N_l(i,j)} \frac{u_{\tilde{i},\tilde{j}} - u_{i,j}}{h_l} \\ 0 &= (a_{i,j}u_{i,j} + b_{i,j}v_{i,j} + c_{i,j})\, b_{i,j} - \alpha \sum_{l \in \{x,y\}} \sum_{\tilde{i},\tilde{j} \in N_l(i,j)} \frac{v_{\tilde{i},\tilde{j}} - v_{i,j}}{h_l} \end{aligned}$$

To set up the iterative scheme, we resolve the equation for $u_{i,j}$ (resp. $v_{i,j}$):

$$\begin{aligned} \left(-a_{i,j}^2 - \alpha \sum_{l \in \{x,y\}} \sum_{\tilde{i},\tilde{j} \in N_l(i,j)} \frac{1}{h_l} \right) u_{i,j} &= a_{i,j}b_{i,j}v_{i,j} + a_{i,j}c_{i,j} - \alpha \sum_{l \in \{x,y\}} \sum_{\tilde{i},\tilde{j} \in N_l(i,j)} \frac{u_{\tilde{i},\tilde{j}}}{h_l} \\ \left(-b_{i,j}^2 - \alpha \sum_{l \in \{x,y\}} \sum_{\tilde{i},\tilde{j} \in N_l(i,j)} \frac{1}{h_l} \right) v_{i,j} &= a_{i,j}b_{i,j}u_{i,j} + b_{i,j}c_{i,j} - \alpha \sum_{l \in \{x,y\}} \sum_{\tilde{i},\tilde{j} \in N_l(i,j)} \frac{v_{\tilde{i},\tilde{j}}}{h_l} \end{aligned}$$

The final iterative scheme then computes the new values $u_{i,j}^{k+1}$ and $v_{i,j}^{k+1}$ based on the variables at the old time step:

$$\begin{aligned} u_{i,j}^{k+1} &= \frac{a_{i,j}b_{i,j}v_{i,j}^k + a_{i,j}c_{i,j} - \alpha \sum\limits_{l \in \{x,y\}} \sum\limits_{\tilde{i},\tilde{j} \in N_l(i,j)} \frac{u^k_{\tilde{i},\tilde{j}}}{h_l}}{\left(-a_{i,j}^2 - \alpha \sum\limits_{l \in \{x,y\}} \sum\limits_{\tilde{i},\tilde{j} \in N_l(i,j)} \frac{1}{h_l} \right)} \\ v_{i,j}^{k+1} &= \frac{a_{i,j}b_{i,j}u_{i,j}^k + b_{i,j}c_{i,j} - \alpha \sum\limits_{l \in \{x,y\}} \sum\limits_{\tilde{i},\tilde{j} \in N_l(i,j)} \frac{v^k_{\tilde{i},\tilde{j}}}{h_l}}{\left(-b_{i,j}^2 - \alpha \sum\limits_{l \in \{x,y\}} \sum\limits_{\tilde{i},\tilde{j} \in N_l(i,j)} \frac{1}{h_l} \right)} \end{aligned}$$

Problem 3.2 (Stereo) 6 Points

The full projection matrix is given by

$$P = \underbrace{\begin{pmatrix} fk_u & -fk_u \cot\theta & u_0 \\ 0 & fk_v/\sin\theta & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{intrinsic matrix } H} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\text{projection matrix}} \underbrace{\begin{pmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & 0 \\ r_{2,1} & r_{2,2} & r_{2,3} & 0 \\ r_{3,1} & r_{3,2} & r_{3,3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\text{extrinsic matrix } T \cdot R}$$

The parameters given in this problem are

$$\begin{aligned} (h_u, h_v) = (k_u^{-1}, k_v^{-1}) &= (1,1) \\ \theta &= 90° \\ (u_0, v_0) &= (2,3) \\ f &= 2 \\ (t_0, t_1, t_2) &= (5, 0, -1) \end{aligned}$$

The rotation matrix is given by

$$\begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Putting all together, we get

$$\begin{aligned} P &= \begin{pmatrix} 2 & 0 & 2 \\ 0 & 2 & 3 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 2 & 0 & 2 \\ 0 & 2 & 3 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 & 5 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 2 & 0 & 2 \\ 0 & 2 & 3 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 & 5 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & -2 & 2 & 8 \\ 2 & 0 & 3 & -3 \\ 0 & 0 & 1 & -1 \end{pmatrix} \end{aligned}$$

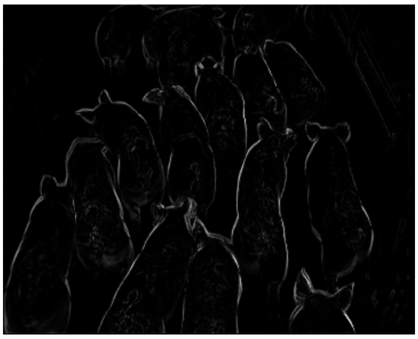
Problem 3.3 (Horn and Schunck) 8 Points

One possible way to set up the solver scheme is:

```
/* Initialize boundaries with zeroes */:
for (i=1; i<=nx; i++)
  {
  v[i][0] = 0;//v[i][1];
  v[i][ny+1] = 0;//v[i][ny];
  }

for (j=0; j<=ny+1; j++)
  {
  v[0][j] = 0;//v[1][j];
  v[nx+1][j] = 0;//v[nx][j];
  }



/* loop over all pixels */
for (i=1; i<=nx; i++)
for (j=1; j<=ny; j++)
  {
  /* Determine size of neighbourhood */
  nn = 4;
  if (i==1 || i==nx) nn--;
  if (j==1 || j==ny) nn--;

  /* Apply iteration step */
  u[i][j] = ( u1[i+1][j] + u1[i-1][j] + u1[i][j+1] + u1[i][j-1]
            - help * fx[i][j] * ( fy[i][j] * v1[i][j] + fz[i][j] ) )
            /
            ( nn + help * fx[i][j] * fx[i][j] );

  v[i][j] = ( v1[i+1][j] + v1[i-1][j] + v1[i][j+1] + v1[i][j-1]
            - help * fy[i][j] * ( fx[i][j] * u1[i][j] + fz[i][j] ) )
            /
            ( nn + help * fy[i][j] * fy[i][j] );
  }
```

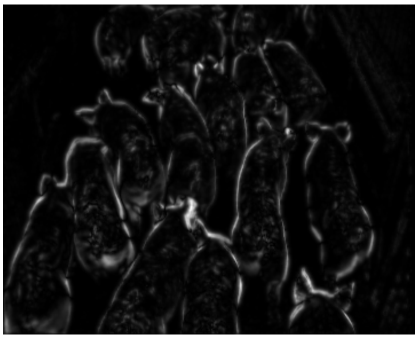
Let us now turn to the experimental evaluation. To this end, we vary the number of iterations and the smoothness weight α in the range $1, 10, 100, 1000$. We can see (next page) that a larger α leads to a stronger filling-in effect (left column), but also that a significant filling-in effect requires sufficiently many iterations to become significant (right column). This is because in every iteration, each pixel communicates only with its four neigbours, thus it needs quite some iterations to propagate through the whole image.

$\alpha = 1, 10$ iterations

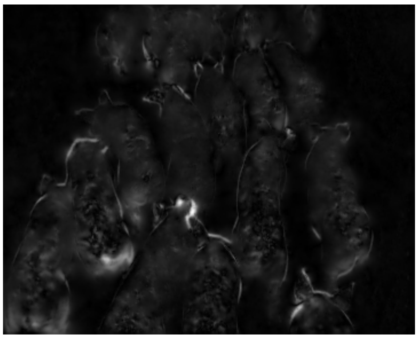


$\alpha = 1000, 1$ iteration

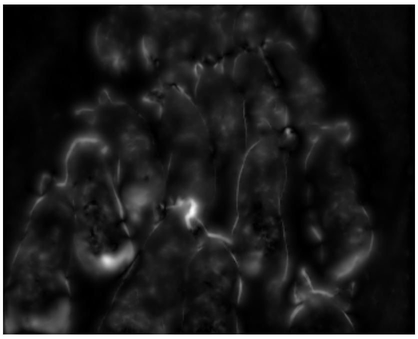
$\alpha = 10, 100$ iterations



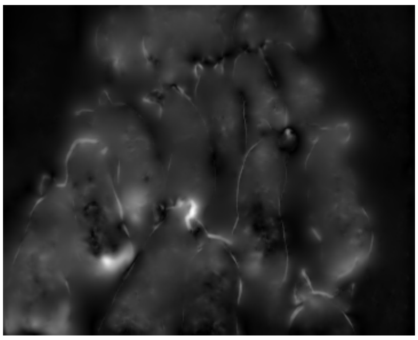
$\alpha = 1000, 10$ iterations



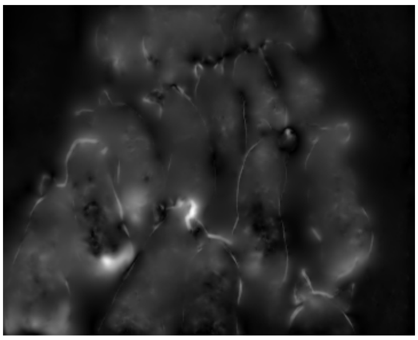
$\alpha = 100, 1000$ iterations



$\alpha = 1000, 100$ iterations



$\alpha = 1000, 1000$ iterations



$\alpha = 1000, 1000$ iterations

Prof. Dr.-Ing. A. Bruhn
Institute for Visualization and Interactive Systems
Department Intelligent Systems
University of Stuttgart

Example Solution for Classroom Assignment 3

C 3.1 (Eigenvalue Analysis)

Since the matrix $J \in \mathbb{R}^{n\times n}$ is symmetric with real components, we know from linear algebra that is has n real eigenvalues $\lambda_1, \ldots, \lambda_n$, and that the corresponding eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ form an orthonormal basis. Moreover, we can decompose J into an orthogonal matrix $Q^\top = (\mathbf{v}_1, \ldots, \mathbf{v}_n)$ and a diagonal matrix $D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, such that $J = Q^\top \, D \, Q$.

Let an arbitrary vector $\mathbf{v}$ with $||\mathbf{v}|| = 1$. The quadratic form then reads

$$E(\mathbf{v}) = \mathbf{v}^\top \; J \; \mathbf{v} = \mathbf{v}^\top \; Q^\top \; D \; Q \; \mathbf{v} = (Q \; \mathbf{v})^\top \;\; D \; Q \; \mathbf{v} \tag{1}$$

Since Q is orthogonal, we have $||Q \; \mathbf{v}|| = ||\mathbf{v}|| = 1$ and thus

$$E(\mathbf{v}) = \mathbf{x}^\top \; D \; \mathbf{x} = \sum_{i=1}^{n} \lambda_i x_i^2. \tag{2}$$

As $x_i^2 > 0$ and $\lambda_i \geq \lambda_1$ for all i, we can write

$$\sum_{i=1}^{n} \lambda_i x_i^2 \geq \sum_{i=1}^{n} \lambda_1 x_i^2. \tag{3}$$

Combining (2) and (3) together with $||Q \; \mathbf{v}|| = 1$ finally yields

$$E(\mathbf{v}) \;=\; \sum_{i=1}^{n} \lambda_i x_i^2 \;\geq\; \sum_{i=1}^{n} \lambda_1 x_i^2 \;=\; \lambda_1 \sum_{i=1}^{n} x_i^2 \;=\; \lambda_1 \tag{4}$$

Thus, the energy is bounded from below by the lowest eigenvalue of J. Since

$$E(\mathbf{v}_1) \;=\; \mathbf{v}_1^\top \; J \; \mathbf{v}_1 \;=\; \mathbf{v}_1^\top \cdot \lambda_1 \mathbf{v}_1 = \lambda_1 ||\mathbf{v}_1|| = \lambda_1, \tag{5}$$

the minimum is attained by the eigenvector that belongs to the lowest eigenvalue.