Some Thoughts in Preparation for Coding Image Processing Problems on a GPU

As a first step we want to consider numerical methods for solving the toy problem,

$$\left\{ \begin{array}{rcll} -\Delta u & = & f, & \Omega \\ u & = & 0, & \partial\Omega \end{array} \right. \tag{1}$$

where $\Omega = (0,1)^d$, and we're particularly interested in the two-dimensional case with $d = 2$. With $x_i = (i - 1/2)h$, $y_j = (j - 1/2)h$, $1 \le i, j \le N$, $h = 1/N$, $N = 2^p$, the following finite difference discretization is natural:

$$\begin{array}{rcl} \frac{1}{h^2}\left[-u(x_{i+1},y_j) + 2u(x_i,y_j) - u(x_{i-1},y_j)\right] & + & \\ \frac{1}{h^2}\left[-u(x_i,y_{j+1}) + 2u(x_i,y_j) - u(x_i,y_{j-1})\right] & = & f(x_i,y_j) \end{array} \tag{2}$$

where terms with $i \pm 1$, $j \pm 1$ outside the range 1 to N are understood to be zero because of the boundary condition $u = 0$, $\partial\Omega$. With the so-called lexicographic ordering,

$$\begin{array}{rcl@{\qquad}rcl} \boldsymbol{x}_k & = & (x_i, y_j) & i & = & \mathrm{mod}(k-1, N) + 1 \\ k & = & (j-1)N + i & j & = & [(k-1)/N] + 1 \end{array} \qquad \text{e.g.,} \quad \begin{array}{|c|c|c|c|} \hline 13 & 14 & 15 & 16 \\ \hline 9 & 10 & 11 & 12 \\ \hline 5 & 6 & 7 & 8 \\ \hline 1 & 2 & 3 & 4 \\ \hline \end{array} \tag{3}$$

the discretization in (2) can be written in terms of $\boldsymbol{u}^{\mathrm{lex}} = \{u_k\}_{k=1}^{N^2}$, $u_k = u(\boldsymbol{x}_k)$ and $\boldsymbol{f}^{\mathrm{lex}} = \{f_k\}_{k=1}^{N^2}$, $f_k = f(\boldsymbol{x}_k)$ as follows:

$$A_{\mathrm{D}}\boldsymbol{u}^{\mathrm{lex}} = \boldsymbol{f}^{\mathrm{lex}} \tag{4}$$

where A_{D} is block tridiagonal:

$$A_{\mathrm{D}} = \frac{1}{h^2}\begin{bmatrix} A_1 & A_0 & & & \\ A_0 & A_1 & A_0 & & \\ & \ddots & \ddots & \ddots & \\ & & A_0 & A_1 & A_0 \\ & & & A_0 & A_1 \end{bmatrix}, \quad A_0 = -I, \quad A_1 = \begin{bmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{bmatrix} \tag{5}$$

and A_0 and A_1 are $N \times N$ blocks.

For the numerical solution of $A_{\mathrm{D}}\boldsymbol{u}^{\mathrm{lex}} = \boldsymbol{f}^{\mathrm{lex}}$ we could begin by considering the simplest Jacobi scheme in which $\sum_{j=1}^{N} a_{ij}u_j = f_i$, $1 \le i \le N$, is rewritten in the following iterative form:

$$(u_i)_{n+1} = \frac{1}{a_{ii}}\left\{ f_i - \sum_{j \ne i} a_{ij}(u_j)_n \right\}, \quad i = 1, \ldots, N \tag{6}$$

Note that the Jacobi scheme has the advantageous property that all components of $\boldsymbol{u}_{n+1}$ can be computed simultaneously without the computation of one component being dependent upon another. This approach requires that $\boldsymbol{u}_{n+1}$ and $\boldsymbol{u}_n$ be stored separately. If the components $\{u_j\}_{j=1}^{i-1}$ are immediately overwritten after computing them, the Jacobi method becomes the Gauss-Seidel method:

$$(u_i)_{n+1} = \frac{1}{a_{ii}}\left\{ f_i - \sum_{j=1}^{i-1} a_{ij}(u_j)_{n+1} - \sum_{j=i+1}^{N} a_{ij}(u_j)_n \right\}, \quad i = 1, \ldots, N \tag{7}$$

This scheme typically has better convergence properties, in the sense that fewer iterations are required to reach a given accuracy. However, the Gauss-Seidel scheme requires that $\{(u_j)_{n+1}\}_{j=1}^{i-1}$

first be calculated before $(u_i)_{n+1}$ can be. This scheme also has an asymmetrical aspect to it, since unknowns are updated from 1 to N. If such a sweep from 1 to N is followed by a sweep from N to 1, then the so-called symmetric Gauss-Seidel scheme is obtained. If the update is weighted by $(1-\omega)$ of the previous iterate and by ω of the new calculation, then symmetric successive overrelaxation method (SSOR) is obtained:

$$\begin{aligned}
(u_i)_{n+\frac{1}{2}} &= (1-\omega)(u_i)_n + \frac{\omega}{a_{ii}}\left\{ f_i - \sum_{j=1}^{i-1} a_{ij}(u_j)_{n+\frac{1}{2}} - \sum_{j=i+1}^{N} a_{ij}(u_j)_n \right\}, \quad i = 1,\ldots,N \\
(u_i)_{n+1} &= (1-\omega)(u_i)_n + \frac{\omega}{a_{ii}}\left\{ f_i - \sum_{j=1}^{i-1} a_{ij}(u_j)_{n+\frac{1}{2}} - \sum_{j=i+1}^{N} a_{ij}(u_j)_{n+1} \right\}, \quad i = N,\ldots,1
\end{aligned} \tag{8}$$

where $\omega \in (0,2)$ is a so-called relaxation parameter. Note in the first equation that the components $(u_j)_{n+\frac{1}{2}}$ for $1 \le j \le i-1$ are immediately overwritten by the most current values obtained by this equation. After the computations of the first equation are completed, the computations of the second equation are carried out. Note similarly in the second equation that the components $(u_j)_{n+1}$ for $i+1 \le j \le N$ are immediately overwritten by the most current values obtained by this equation. These formulas can also be written more compactly in terms of the splitting of A into its strictly diagonal part D, its stictly lower triangular part L and its strictly upper triangular part $U = L^{\mathrm{T}}$,

$$A_{\mathrm{D}} = D + L + L^{\mathrm{T}} \tag{9}$$

according to:

$$\begin{aligned}
\boldsymbol{u}^{\mathrm{lex}}_{n+\frac{1}{2}} &= (1-\omega)\boldsymbol{u}^{\mathrm{lex}}_{n} + \omega D^{-1}\left\{ \boldsymbol{f} - L\boldsymbol{u}^{\mathrm{lex}}_{n+\frac{1}{2}} - L^{\mathrm{T}}\boldsymbol{u}^{\mathrm{lex}}_{n} \right\} \\
\boldsymbol{u}^{\mathrm{lex}}_{n+1} &= (1-\omega)\boldsymbol{u}^{\mathrm{lex}}_{n} + \omega D^{-1}\left\{ \boldsymbol{f} - L\boldsymbol{u}^{\mathrm{lex}}_{n+\frac{1}{2}} - L^{\mathrm{T}}\boldsymbol{u}^{\mathrm{lex}}_{n+1} \right\}
\end{aligned} \tag{10}$$

or after solving for $\boldsymbol{u}^{\mathrm{lex}}_{n+1/2}$ and $\boldsymbol{u}^{\mathrm{lex}}_{n+1}$ respectively:

$$\begin{aligned}
\boldsymbol{u}^{\mathrm{lex}}_{n+\frac{1}{2}} &= (D+\omega L)^{-1}\{[(1-\omega)D - \omega L^{\mathrm{T}}]\boldsymbol{u}^{\mathrm{lex}}_{n} + \omega \boldsymbol{f}^{\mathrm{lex}}\} \\
&= \boldsymbol{u}^{\mathrm{lex}}_{n} + \omega(D+\omega L)^{-1}[\boldsymbol{f}^{\mathrm{lex}} - A_{\mathrm{D}}\boldsymbol{u}^{\mathrm{lex}}_{n}] \\
\boldsymbol{u}^{\mathrm{lex}}_{n+1} &= (D+\omega L^{\mathrm{T}})^{-1}\{[(1-\omega)D - \omega L]\boldsymbol{u}^{\mathrm{lex}}_{n+\frac{1}{2}} + \omega \boldsymbol{f}^{\mathrm{lex}}\} \\
&= \boldsymbol{u}^{\mathrm{lex}}_{n+\frac{1}{2}} + \omega(D+\omega L^{\mathrm{T}})^{-1}[\boldsymbol{f}^{\mathrm{lex}} - A_{\mathrm{D}}\boldsymbol{u}^{\mathrm{lex}}_{n+\frac{1}{2}}]
\end{aligned} \tag{11}$$

From (8) it is apparent that the unknowns of $\boldsymbol{u}^{\mathrm{lex}}$ must be updated in a sequential fashion using the lexicographic ordering. On the other hand, a degree of parallelism can be achieved with the following red-black (in general, multicolor) ordering:

$$\boldsymbol{x}_l = \boldsymbol{x}_{k(l)}, \qquad k(l) = \begin{cases} 2l-1, & l = 1,\ldots,N^2/2 \\ 2l-N^2, & l = N^2/2+1,\ldots,N \end{cases} \qquad \text{e.g.,} \quad \begin{array}{|c|c|c|c|} \hline 15 & 7 & 16 & 8 \\ \hline 5 & 13 & 6 & 14 \\ \hline 11 & 3 & 12 & 4 \\ \hline 1 & 9 & 2 & 10 \\ \hline \end{array} \tag{12}$$

where $\boldsymbol{x}_k$ is given by (3). Now with $\boldsymbol{u}^{\mathrm{rb}} = \{u_l\}_{k=1}^{N^2}$, $u_l = u(\boldsymbol{x}_l)$ and $\boldsymbol{f}^{\mathrm{rb}} = \{f_l\}_{l=1}^{N^2}$, $f_l = f(\boldsymbol{x}_l)$, (1) is discretized as follows:

$$B_{\mathrm{D}}\boldsymbol{u}^{\mathrm{rb}} = \boldsymbol{f}^{\mathrm{rb}} \tag{13}$$

where

$$B_{\mathrm{D}} = \frac{1}{h^2}\left[\begin{array}{ccccc|ccccc} B_2 & & & & & B_1 & B_0 & & & \\ & B_2 & & & & B_0 & B_1^{\mathrm{T}} & B_0 & & \\ & & \ddots & & & & \ddots & \ddots & \ddots & \\ & & & B_2 & & & & B_0 & B_1 & B_0 \\ & & & & B_2 & & & & B_0 & B_1^{\mathrm{T}} \\ \hline B_1^{\mathrm{T}} & B_0 & & & & B_2 & & & & \\ B_0 & B_1 & B_0 & & & & B_2 & & & \\ & \ddots & \ddots & \ddots & & & & \ddots & & \\ & & B_0 & B_1^{\mathrm{T}} & B_0 & & & & B_2 & \\ & & & B_0 & B_1 & & & & & B_2 \end{array}\right]$$

$$B_2 = 4I, \quad B_0 = -I,$$

$$B_1 = \left[\begin{array}{ccccc} -1 & & & & \\ -1 & -1 & & & \\ & \ddots & \ddots & & \\ & & -1 & -1 & \\ & & & -1 & -1 \end{array}\right]$$

(14)

and B_0, B_1 and B_2 are $\frac{N}{2} \times \frac{N}{2}$ blocks. With the red-black ordering, the SSOR method takes a particularly simple form. For this let $\boldsymbol{u}^{\mathrm{rb}}$ and $\boldsymbol{f}^{\mathrm{rb}}$ be partitioned into red and black components:

$$\boldsymbol{u}_{\mathrm{R}} = \{u_l\}_{l=1}^{N^2/2}, \quad \boldsymbol{u}_{\mathrm{B}} = \{u_l\}_{l=N^2/2+1}^{N^2}, \qquad \boldsymbol{f}_{\mathrm{R}} = \{f_l\}_{l=1}^{N^2/2}, \quad \boldsymbol{f}_{\mathrm{B}} = \{f_l\}_{l=N^2/2+1}^{N^2} \tag{15}$$

Also let B_{D} be partitioned correspondingly,

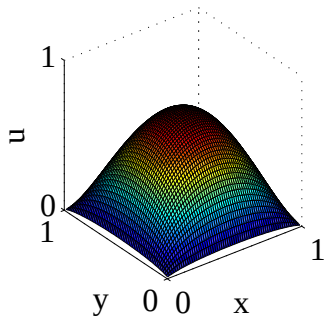
$$B_{\mathrm{D}} = \left[\begin{array}{c|c} D_0 & L_0^{\mathrm{T}} \\ \hline L_0 & D_0 \end{array}\right] \tag{16}$$

Then the SSOR method can be written as:

$$\begin{aligned} (\boldsymbol{u}_{\mathrm{R}})_{n+\frac{1}{2}} &= (1-\omega)(\boldsymbol{u}_{\mathrm{R}})_n + \omega D_0^{-1}[\boldsymbol{f}_{\mathrm{R}} - L_0^{\mathrm{T}}(\boldsymbol{u}_{\mathrm{B}})_n] \\ (\boldsymbol{u}_{\mathrm{B}})_{n+\frac{1}{2}} &= (1-\omega)(\boldsymbol{u}_{\mathrm{B}})_n + \omega D_0^{-1}[\boldsymbol{f}_{\mathrm{B}} - L_0(\boldsymbol{u}_{\mathrm{R}})_{n+\frac{1}{2}}] \\ (\boldsymbol{u}_{\mathrm{B}})_{n+1} &= (1-\omega)(\boldsymbol{u}_{\mathrm{B}})_{n+\frac{1}{2}} + \omega D_0^{-1}[\boldsymbol{f}_{\mathrm{B}} - L_0(\boldsymbol{u}_{\mathrm{R}})_{n+\frac{1}{2}}] \\ (\boldsymbol{u}_{\mathrm{R}})_{n+1} &= (1-\omega)(\boldsymbol{u}_{\mathrm{R}})_{n+\frac{1}{2}} + \omega D_0^{-1}[\boldsymbol{f}_{\mathrm{R}} - L_0^{\mathrm{T}}(\boldsymbol{u}_{\mathrm{B}})_{n+1}] \end{aligned} \tag{17}$$

where all elements of $\boldsymbol{u}_{\mathrm{R}}$ are updated simultaneously and then all elements of $\boldsymbol{u}_{\mathrm{B}}$ are updated simultaneously.

An example of a solution to (1) is shown in the following figure for $f = 10$.



Since the nonlinear problems analyzed below involve boundary conditions different than seen in (1) we consider first the following variation of the linear problem:

$$\left\{\begin{array}{rcll} -\mu\Delta u + u &=& f, & \Omega \\ \partial u/\partial n &=& 0, & \partial\Omega \end{array}\right. \tag{18}$$

where the so-called homogeneous Dirichlet boundary condition is given in (1) and the homogeneous Neumann boundary condition is given in (18). Note that without the zeroth-order term u in the operator $-\mu\Delta u + u$, no unique solution would exist, since any constant could be added to a potential solution u and the sum would still satisfy the equation. For (18), the matrix formulation in (5) must be modified as follows:

$$A_{\mathrm{N}} = \frac{\mu}{h^2}\begin{bmatrix} A_1 & A_0 & & & & & & \\ A_0 & A_2 & A_0 & & & & & \\ & A_0 & A_3 & A_0 & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & A_0 & A_3 & A_0 & & \\ & & & & A_0 & A_2 & A_0 & \\ & & & & & A_0 & A_1 \end{bmatrix} + I \tag{19}$$

$$A_0 = -I, \quad A_1 = \begin{bmatrix} 2 & -1 & & & \\ -1 & 3 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 3 & -1 \\ & & & -1 & 2 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 3 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 3 \end{bmatrix}$$

and A_0, A_1 and A_2 are $N \times N$ blocks. This discretization can be conceived by constructing a layer of ghost cells just outside of Ω, say with centers (x_κ, y_λ), where at least one of the cases $\{\kappa = 0 \text{ or } N+1,\ \lambda = 0 \text{ or } N+1\}$ hold. Let (x_k, y_l) be the center of the cell within Ω which is nearest to (x_κ, y_λ). Then $u_{\kappa,\lambda} = u_{k,l}$ must hold because of the Neumann boundary condition $u_{\mathrm{n}} = 0$. So the usual stencil, $h^2 u_{xx}(x_1, y_1) \approx u_{2,1} - 2u_{1,1} + u_{0,1} = [u_{2,1} - u_{1,1}] - [u_{1,1} - u_{0,1}]$ becomes $u_{2,1} - u_{1,1}$ since $u_{1,1} - u_{0,1} = 0$ holds in terms of the ghost cell value $u_{0,1}$.

To obtain the discretization of (18) with red-black ordering, the matrix formulation in (14) must be modified as follows:

$$B_{\mathrm{N}} = \frac{\mu}{h^2}\left[\begin{array}{ccccccc|ccccccc} B_2 & & & & & & & B_1 & B_0 & & & & & \\ & B_3 & & & & & & B_0 & B_1^{\mathrm{T}} & B_0 & & & & \\ & & B_4 & & & & & & B_0 & B_1 & B_0 & & & \\ & & & \ddots & & & & & & \ddots & \ddots & \ddots & & \\ & & & & B_3 & & & & & & B_0 & B_1^{\mathrm{T}} & B_0 & \\ & & & & & B_4 & & & & & & B_0 & B_1 & B_0 \\ & & & & & & B_5 & & & & & & B_0 & B_1^{\mathrm{T}} \\ \hline B_1^{\mathrm{T}} & B_0 & & & & & & B_5 & & & & & & \\ B_0 & B_1 & B_0 & & & & & & B_4 & & & & & \\ & B_0 & B_1^{\mathrm{T}} & B_0 & & & & & & B_3 & & & & \\ & & \ddots & \ddots & \ddots & & & & & & \ddots & & & \\ & & & B_0 & B_1 & B_0 & & & & & & B_4 & & \\ & & & & B_0 & B_1^{\mathrm{T}} & B_0 & & & & & & B_3 & \\ & & & & & B_0 & B_1 & & & & & & & B_2 \end{array}\right] + I \tag{20}$$

$$B_2 = \begin{bmatrix} 2 & & & \\ & 3 & & \\ & & \ddots & \\ & & & 3 \end{bmatrix}, \quad B_3 = \begin{bmatrix} 4 & & & \\ & \ddots & & \\ & & 4 & \\ & & & 3 \end{bmatrix}, \quad B_4 = \begin{bmatrix} 3 & & & \\ & 4 & & \\ & & \ddots & \\ & & & 4 \end{bmatrix}$$

$$B_5 = \begin{bmatrix} 3 & & & \\ & \ddots & & \\ & & 3 & \\ & & & 2 \end{bmatrix} \quad B_1 = \begin{bmatrix} -1 & & & & \\ -1 & -1 & & & \\ & \ddots & \ddots & & \\ & & -1 & -1 & \\ & & & -1 & -1 \end{bmatrix} \quad B_0 = -I \tag{21}$$

where B_0, B_1, B_2, B_3, B_4 and B_5 are $\frac{N}{2} \times \frac{N}{2}$ blocks. Again, B_{N} has the same partitioning as in (16),

$$B_{\mathrm{N}} = \left[\begin{array}{c|c} D_0 & L_0^{\mathrm{T}} \\ \hline L_0 & D_0 \end{array}\right] \qquad D = \left[\begin{array}{c|c} D_0 & \\ \hline & D_0 \end{array}\right] \tag{22}$$

and, by writing $(1-\omega)(\boldsymbol{u}_{\mathrm{R}})_n + \omega D_0^{-1}[\boldsymbol{f}_{\mathrm{R}} - L_0^{\mathrm{T}}(\boldsymbol{u}_{\mathrm{B}})_n]$ in (17) as $(\boldsymbol{u}_{\mathrm{R}})_n + \omega D_0^{-1}[\boldsymbol{f}_{\mathrm{B}} - D_0(\boldsymbol{u}_{\mathrm{R}})_n - L_0^{\mathrm{T}}(\boldsymbol{u}_{\mathrm{B}})_n]$ or equivalently as $(\boldsymbol{u}_{\mathrm{R}})_n + \omega D_0^{-1}[\boldsymbol{f} - B_{\mathrm{D}}(\boldsymbol{u})_n]_{\mathrm{B}}$ the first step of the SSOR method applied for B_{N} can be written similarly in the form:

$$P_{\mathrm{R}}\boldsymbol{u}_{n+\frac{1}{2}} = P_{\mathrm{R}}\boldsymbol{u}_n + \omega P_{\mathrm{R}} D^{-1} P_{\mathrm{R}}[\boldsymbol{f} - B_{\mathrm{N}}\boldsymbol{u}_n] \tag{23}$$

where P_{R} and P_{B} are projections onto red and black cells, respectively, i.e., P_{R} is 1 on red cells and zero on black cells and $P_{\mathrm{B}} = 1 - P_{\mathrm{R}}$. Note that with these projections, B_{N} could even be replaced by A_{N} in (23), provided D is replaced by the diagonal of A_{N}. This rather mobile structure of (23) suggests to formulate the iteration independently of the matrix structure. This can be done using stencil formulas as follows.

Now let $\boldsymbol{u} = \{u_{ij}\}$ denote the two-dimensional array of cell values where $u_{ij} \approx u(x_i, y_j)$. Then let a differential operator be approximated by a discrete operator,

$$L\boldsymbol{u} = \sum_{|k|,|l| \le r} \boldsymbol{\sigma}_{k,l} S_{k,l} \boldsymbol{u} \tag{24}$$

where $S_{k,l}$ is a shift operator which wraps at the boundary as illustrated below in (30),

$$(S_{k,l}\boldsymbol{u})_{i,j} = u_{\lfloor i+k \rceil, \lfloor j+l \rceil}, \quad \lfloor i+k \rceil = \begin{cases} i+k+N, & i+k < 1 \\ i+k, & 1 \le i+k \le N \\ i+k-N, & N < i+k \end{cases} \tag{25}$$

Also the stencil at each cell (x_i, y_j) consists of weights $\{(\sigma_{i,j})_{k,l}\}_{|k|,|l| \le r}$ in the sum of neighboring cells $\{(x_{i+k}, y_{j+l}) : |k|, |l| \le r\}$ in (24). For instance, the weights corresponding to (2) at non-boundary cells are $\boldsymbol{\sigma}_{\pm 1,0} = \boldsymbol{\sigma}_{0,\pm 1} = -1/h^2$ and $\boldsymbol{\sigma}_{0,0} = +4/h^2$. These weights must be modified at the boundary depending upon whether Dirichlet or Neumann boundary conditions are used. Specifically, for Dirichlet boundary conditions the stencils at the corner cells $\{(x_i, y_j) : 1 \le i, j, \le 2\}$ are obtained by dividing the following by h^2:

(x_1, y_2) :

0	−1	0
0	4	−1
0	−1	0

(x_2, y_2) :

0	−1	0
−1	4	−1
0	−1	0

(x_1, y_1) :

0	−1	0
0	4	−1
0	0	0

(x_2, y_1) :

0	−1	0
−1	4	−1
0	0	0

(26)

and the stencils for other boundary cells are given similarly. For Neumann boundary conditions the stencils at the corner cells $\{(x_i, y_j) : 1 \le i, j, \le 2\}$ are obtained by dividing the following by h^2:

(x_1, y_2) :

0	−1	0
0	3	−1
0	−1	0

(x_2, y_2) :

0	−1	0
−1	4	−1
0	−1	0

(x_1, y_1) :

0	−1	0
0	2	−1
0	0	0

(x_2, y_1) :

0	−1	0
−1	3	−1
0	0	0

(27)

and the stencils for other boundary cells are given similarly.

With the Dirichlet or Neumann problem so formulated, (23) can be written as:

$$P_{\mathrm{R}}\boldsymbol{u}_{n+\frac{1}{2}} = P_{\mathrm{R}}\boldsymbol{u}_n + \omega P_{\mathrm{R}}\boldsymbol{\sigma}_{0,0}^{-1}P_{\mathrm{R}}[\boldsymbol{f} - L\boldsymbol{u}_n] \tag{28}$$

Note that $P_{\mathrm{R}}L\boldsymbol{u}$ can be written as:

$$P_{\mathrm{R}}L\boldsymbol{u} = \sum_{|k|+|l|\le 1} P_{\mathrm{R}}\boldsymbol{\sigma}_{k,l}P_{\mathrm{R}}S_{k,l}\boldsymbol{u} \tag{29}$$

Instead of processing many zero values resulting from projections in such an equation, we define the index set $I_{\mathrm{R}}^{k,l}$ so that $\boldsymbol{u}(I_{\mathrm{R}}^{k,l})$ is the set of nontrivial red cell values of $P_{\mathrm{R}}S_{k,l}\boldsymbol{u}$. In particular, let $I_{\mathrm{R}} = I_{\mathrm{R}}^{0,0}$ so that $\boldsymbol{u}(I_{\mathrm{R}})$ is the set of nontrivial red cell values in $P_{\mathrm{R}}\boldsymbol{u}$. Explicitly, with respect to the sample grid of (12), $\{I_{\mathrm{R}}^{\pm1,\pm1}\}$ are the index sets for the following sets of cells:

$I_{\mathrm{R}}^{-1,0}$:

	(1,4)	R	(3,4)	R
(4,3)	R	(2,3)	R	
	(1,2)	R	(3,2)	R
(4,1)	R	(2,1)	R	

$I_{\mathrm{R}}^{+1,0}$:

	R	(3,4)	R	(1,4)
R	(2,3)	R	(4,3)	
	R	(3,2)	R	(1,2)
R	(2,1)	R	(4,1)	

$I_{\mathrm{R}}^{0,-1}$:

	R		R
R	(2,3)	R	(4,3)
(1,2)	R	(3,2)	R
R	(2,1)	R	(4,1)
(1,4)		(3,4)	

$I_{\mathrm{R}}^{0,+1}$:

	(2,1)		(4,1)
(1,4)	R	(3,4)	R
R	(2,3)	R	(4,3)
(1,2)	R	(3,2)	R
R		R	

(30)

where the red cells above correspond to the index set,

$I_{\mathrm{R}}^{0,0} = I_{\mathrm{R}}$:

	(2,4)		(4,4)
(1,3)		(3,3)	
	(2,2)		(4,2)
(1,1)		(3,1)	

(31)

Define $I_{\mathrm{B}}^{k,l}$ and $I_{\mathrm{B}} = I_{\mathrm{B}}^{0,0}$ similarly with respect to black cell values. With these constructions, (29) can be written as:

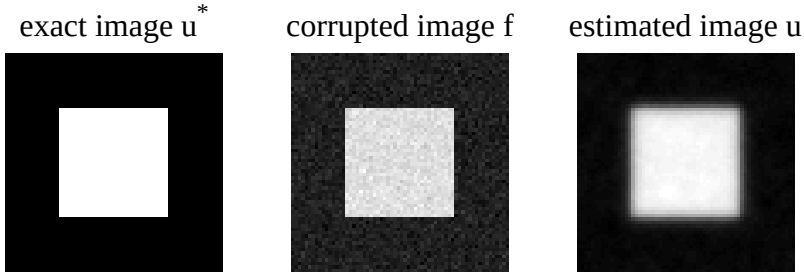
$$(Lu)(I_{\mathrm{R}}) = \sum_{|k|+|l|\le 1} \boldsymbol{\sigma}_{k,l}(I_{\mathrm{R}})\boldsymbol{u}(I_{\mathrm{R}}^{k,l}) \tag{32}$$

and so (28) can be written as:

$$\boldsymbol{u}_{n+\frac{1}{2}}(I_{\mathrm{R}}) = \boldsymbol{u}_n(I_{\mathrm{R}}) + \frac{\omega}{\boldsymbol{\sigma}_{0,0}(I_{\mathrm{R}})}\left[\boldsymbol{f}(I_{\mathrm{R}}) - \sum_{|k|+|l|\le 1} \boldsymbol{\sigma}_{k,l}(I_{\mathrm{R}})\boldsymbol{u}(I_{\mathrm{R}}^{k,l})\right] \tag{33}$$

with the terms corresponding to those remaining in (17) written similarly.

An example of a solution u to (18) is shown on the right in the following figure next to (a noisy image) f shown in the middle and an ideal image u^* shown on the left.



This problem is a special case of minimizing the functional in (34) below for no blurring $\mathrm{K} = I$ and for regularization $\phi(s) = \mu s$. The result can be improved for other choices of ϕ.

So what we really want to do is to minimize functionals such as the following,

$$J(u) = \int_\Omega [\mathrm{K}u - \tilde{u}]^2 d\boldsymbol{x} + \int_\Omega \phi(|\nabla u|^2) d\boldsymbol{x} \tag{34}$$

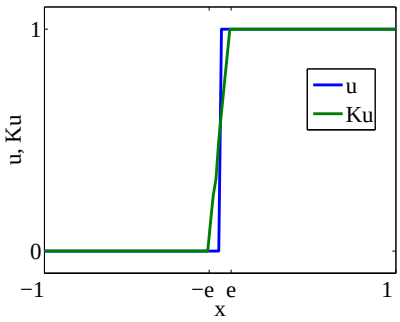
where $\tilde{u}$ is a given noisy and blurred image. Blurring is modeled by the convolution operator K,

$$[\mathrm{K}u](\boldsymbol{x}) = \int_\Omega \kappa(\boldsymbol{x} - \boldsymbol{y}) u(\boldsymbol{y}) d\boldsymbol{y} \tag{35}$$

defined in terms of a so-called point-spread function κ such as

$$\kappa(\boldsymbol{z}) = \begin{cases} (2\epsilon)^{-d}, & |\boldsymbol{z}| < \epsilon \\ 0, & \text{else} \end{cases} \qquad (d = 2) \tag{36}$$

For instance, note the following effect of the operator K in the one-dimensional case when u is a step function:



Specifically, K smoothes the step function to be linear between $-\epsilon$ and ϵ. Applying K again would create a function $\mathrm{K}^2 u$ which is smoothly quadratic between -2ϵ and 2ϵ. In other words, K has a smoothing effect which leads to a blur of $\mathrm{K}u$ in relation to u. We assume also that $\kappa(z) = \kappa(-z)$ holds, so the operator K is symmetric,

$$\int_\Omega v\mathrm{K}u d\boldsymbol{x} = \int_\Omega u\mathrm{K}v d\boldsymbol{x}. \tag{37}$$

Possible choices for the penalty function ϕ are $\phi(s) = \mu s^{p/2}$, $1 \le p \le 2$, where edges are particularly well captured by $p = 1$ and gradual slopes are better captured by $p = 2$. Thus, a

function ϕ which is closer to the case $p = 1$ when s is large ($|\nabla u|^2$ is large) and closer to the case $p = 2$ when s is small ($|\nabla u|^2$ is small) is ideal. Specifically, we might take ϕ according to:

$$\phi_\epsilon(s) = \mu \begin{cases} s/(2\sqrt{\epsilon}), & 0 \le s \le \epsilon \\ \sqrt{s} - \sqrt{\epsilon}/2, & s \ge \epsilon \end{cases} \qquad \phi'_\epsilon(s) = \frac{\mu}{2\max\{\epsilon, s\}} \tag{38}$$

The minimizer u of J must satisfy the necessary optimality condition with respect to any sufficiently smooth perturbation $\bar{u}$,

$$\begin{aligned} 0 = \frac{1}{2}\frac{\delta J}{\delta u}(u;\bar{u}) = \frac{1}{2}\lim_{\epsilon\to 0}\frac{d}{d\epsilon}J(u+\epsilon\bar{u}) = \int_\Omega [\mathrm{K}u - \tilde{u}]\mathrm{K}\bar{u}d\boldsymbol{x} + \int_\Omega \phi'(|\nabla u|^2)(\nabla u\cdot\nabla\bar{u})d\boldsymbol{x} \\ = \int_\Omega [\mathrm{K}^2u - \mathrm{K}\tilde{u}]\bar{u}d\boldsymbol{x} + \int_{\partial\Omega}\bar{u}\phi'(|\nabla u|^2)(\nabla u\cdot\boldsymbol{n})dS(\boldsymbol{x}) - \int_\Omega \bar{u}\nabla\cdot[\phi'(|\nabla u|^2)\nabla u]d\boldsymbol{x} \end{aligned} \tag{39}$$

Note that the last two integrals emerge from partial integration. Also, $\boldsymbol{n}$ denotes the unit normal vector which is outwardly directed at $\partial\Omega$. So the normal derivative of u at $\partial\Omega$ is given by $\partial u/\partial n = \nabla u\cdot\boldsymbol{n}$. The directional derivatives of J are now given succinctly by:

$$0 = \frac{1}{2}\frac{\delta J}{\delta u}(u;\bar{u}) = \int_\Omega \bar{u}\left\{\mathrm{K}^2u - \mathrm{K}\tilde{u} - \nabla\cdot[\phi'(|\nabla u|^2)\nabla u]\right\}d\boldsymbol{x} + \int_{\partial\Omega}\bar{u}\phi'(|\nabla u|^2)\frac{\partial u}{\partial n}dS(\boldsymbol{x}) \tag{40}$$

Since these directional derivatives must be zero for all sufficiently smooth perturbations $\bar{u}$, letting the perturbation $\bar{u}$ be concentrated at points in Ω or on $\partial\Omega$ leads to the following pointwise characterization of the minimizer u:

$$\begin{cases} -\nabla\cdot[\phi'(|\nabla u|^2)\nabla u] + \mathrm{K}^2u &= \mathrm{K}\tilde{u}, & \Omega \\ \partial u/\partial n &= 0, & \partial\Omega \end{cases} \tag{41}$$

Since this problem is nonlinear we apply Newton's method. In the function space setting, Newton's method takes the form:

$$\frac{\delta^2 J}{\delta u^2}(u_m;\bar{u},\delta u) = -\frac{\delta J}{\delta u}(u_m;\bar{u}), \quad u_{m+1} = u_m + \alpha\delta u \tag{42}$$

where α may be chosen by a line search to minimize $f(\alpha) = J(u+\alpha\delta u)$, and the second variation of J is given by:

$$\begin{aligned} \frac{1}{2}\frac{\delta^2 J}{\delta u^2}(u;\bar{u}_1,\bar{u}_2) = \frac{1}{2}\lim_{\epsilon\to 0}\frac{d}{d\epsilon}\frac{\delta J}{\delta u}(u+\epsilon\bar{u}_2;\bar{u}_1) = \\ \int_\Omega \mathrm{K}\bar{u}_1\mathrm{K}\bar{u}_2 d\boldsymbol{x} + \int_\Omega\left[\phi'(|\nabla u|^2)(\nabla\bar{u}_1\cdot\nabla\bar{u}_2) + \phi''(\nabla u\cdot\nabla\bar{u}_1)(\nabla u\cdot\nabla\bar{u}_2)\right]d\boldsymbol{x} \end{aligned} \tag{43}$$

In order to guarantee the positivity of the second derivative operator to be inverted, we neglect the term with ϕ'' whose positivity is not assured; see, e.g., (38). Thus, the second variation of J is approximated according to:

$$\frac{1}{2}\frac{\delta^2 J}{\delta u^2}(u;\bar{u},\delta u) \approx \int_\Omega \bar{u}\left[\mathrm{K}^2\delta u - \nabla\cdot[\phi'(|\nabla u|^2)\nabla\delta u]\right]d\boldsymbol{x} + \int_{\partial\Omega}\bar{u}\phi'(|\nabla u|^2)\frac{\partial\delta u}{\partial n}dS(\boldsymbol{x}) \tag{44}$$

As with (40) and (41), letting the perturbation $\bar{u}$ in (44) be concentrated at points in Ω or on $\partial\Omega$ leads to the following pointwise characterization of the Newton update δu defined in (42):

$$\begin{cases} -\nabla\cdot[\phi'(|\nabla u_m|^2)\nabla\delta u] + \mathrm{K}^2\delta u &= \nabla\cdot[\phi'(|\nabla u_m|^2)\nabla u_m] - \mathrm{K}^2u_m + \mathrm{K}\tilde{u}, & \Omega \\ \partial\delta u/\partial n &= 0, & \partial\Omega \\ u_{m+1} &= u_m + \alpha\delta u, \quad m = 0,1,2,\ldots & \\ u_0 &= \tilde{u} & \end{cases} \tag{45}$$

In analogy with (2), $\nabla \cdot [\phi'(|\nabla u|^2)\nabla v]$ is discretized as follows:

$$\begin{aligned} &\frac{1}{h}\left\{\phi'_{i+\frac{1}{2},j}\left[\frac{v_{i+1,j}-v_{i,j}}{h}\right]-\phi'_{i-\frac{1}{2},j}\left[\frac{v_{i,j}-v_{i-1,j}}{h}\right]\right\} \quad + \\ &\frac{1}{h}\left\{\phi'_{i,j+\frac{1}{2}}\left[\frac{v_{i,j+1}-v_{i,j}}{h}\right]-\phi'_{i,j-\frac{1}{2}}\left[\frac{v_{i,j}-v_{i,j-1}}{h}\right]\right\} \quad \approx \quad \nabla \cdot [\phi'(|\nabla u|^2)\nabla v](x_i,y_j) \end{aligned} \tag{46}$$

where:

$$\phi'_{i\pm\frac{1}{2},j} = \phi'\left(\left|\frac{u_{i\pm1,j}-u_{i,j}}{h}\right|^2+\left|\frac{u_{i,j+1}-u_{i,j-1}+u_{i+1,j+1}-u_{i+1,j-1}}{4h}\right|^2\right) \tag{47}$$

and:

$$\phi'_{i,j\pm\frac{1}{2}} = \phi'\left(\left|\frac{u_{i,j\pm1}-u_{i,j}}{h}\right|^2+\left|\frac{u_{i+1,j}-u_{i-1,j}+u_{i+1,j+1}-u_{i-1,j+1}}{4h}\right|^2\right) \tag{48}$$

Note that the first difference quotient in (47) approximates u_x at $(x_{i\pm\frac{1}{2}}, y_j)$ and the second difference quotient approximates u_y at $(x_{i\pm\frac{1}{2}}, y_j)$. Then in (48) the first difference quotient approximates u_y at $(x_i, y_{j\pm\frac{1}{2}})$ and the second difference quotient approximates u_x at $(x_i, y_{j\pm\frac{1}{2}})$.

Writing this discretization in the lexicographic ordering as in (19) leads to:

$$A(\boldsymbol{u}) = \frac{1}{h^2}\begin{bmatrix} D_1 & L_1 & & & & & \\ L_1 & D_2 & L_2 & & & & \\ & L_2 & D_3 & L_3 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & L_{N-3} & D_{N-2} & L_{N-2} & \\ & & & & L_{N-2} & D_{N-1} & L_{N-1} \\ & & & & & L_{N-1} & D_N \end{bmatrix} \tag{49}$$

where:

$$L_j = -\begin{bmatrix} \phi'_{1,j+\frac{1}{2}} & & & & & \\ & \phi'_{2,j+\frac{1}{2}} & & & & \\ & & \ddots & & & \\ & & & \phi'_{N-1,j+\frac{1}{2}} & & \\ & & & & \phi'_{N,j+\frac{1}{2}} \end{bmatrix} \tag{50}$$

$$D_1 = \begin{bmatrix} \phi'_{1,1+\frac{1}{2}}+\phi'_{1+\frac{1}{2},1} & -\phi'_{1+\frac{1}{2},1} & & & \\ -\phi'_{1+\frac{1}{2},1} & \phi'_{2\pm\frac{1}{2},1}+\phi'_{1,1+\frac{1}{2}} & -\phi'_{2+\frac{1}{2},1} & & \\ & \ddots & \ddots & \ddots & \\ & & -\phi'_{N-1-\frac{1}{2},1} & \phi'_{N-1\pm\frac{1}{2},1}+\phi'_{N-1,1+\frac{1}{2}} & -\phi'_{N-\frac{1}{2},1} \\ & & & -\phi'_{N-\frac{1}{2},1} & \phi'_{N,1+\frac{1}{2}}+\phi'_{N-\frac{1}{2},1} \end{bmatrix} \tag{51}$$

$$D_N = \begin{bmatrix} \phi'_{1,N-\frac{1}{2}}+\phi'_{1+\frac{1}{2},N} & -\phi'_{1+\frac{1}{2},N} & & & \\ -\phi'_{1+\frac{1}{2},N} & \phi'_{2\pm\frac{1}{2},N}+\phi'_{1,N-\frac{1}{2}} & -\phi'_{2+\frac{1}{2},N} & & \\ & \ddots & \ddots & \ddots & \\ & & -\phi'_{N-1-\frac{1}{2},N} & \phi'_{N-1\pm\frac{1}{2},N}+\phi'_{N-1,N-\frac{1}{2}} & -\phi'_{N-\frac{1}{2},N} \\ & & & -\phi'_{N-\frac{1}{2},1} & \phi'_{N,N-\frac{1}{2}}+\phi'_{N-\frac{1}{2},N} \end{bmatrix} \tag{52}$$

and for $1 < j < N$,

$$D_j = \begin{bmatrix} \phi'_{1,j\pm\frac{1}{2}}+\phi'_{1+\frac{1}{2},j} & -\phi'_{1+\frac{1}{2},j} & & & \\ -\phi'_{1+\frac{1}{2},j} & \phi'_{2\pm\frac{1}{2},j}+\phi'_{1,j\pm\frac{1}{2}} & -\phi'_{2+\frac{1}{2},j} & & \\ & \ddots & \ddots & \ddots & \\ & & -\phi'_{N-1-\frac{1}{2},j} & \phi'_{N-1\pm\frac{1}{2},j}+\phi'_{N-1,j\pm\frac{1}{2}} & -\phi'_{N-\frac{1}{2},j} \\ & & & -\phi'_{N-\frac{1}{2},j} & \phi'_{N,j\pm\frac{1}{2}}+\phi'_{N-\frac{1}{2},j} \end{bmatrix} \tag{53}$$

Note that in these formulas for D_1, D_j and D_N as well as in the stencils below involving ϕ', $\phi'_{i,j\pm1/2}$, for instance, denotes the sum of $\phi'_{i,j+1/2}$ and $\phi'_{i,j-1/2}$. The convolution operator can be discretized in the lexicographic ordering as:

$$K = \frac{1}{8}\begin{bmatrix} K_1 & K_0 & & & & & & \\ K_0 & K_2 & K_0 & & & & & \\ & K_0 & K_2 & K_0 & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & K_0 & K_2 & K_0 & & \\ & & & & K_0 & K_2 & K_0 & \\ & & & & & K_0 & K_1 \end{bmatrix} \tag{54}$$

$$K_0 = I, \quad K_1 = \begin{bmatrix} 6 & 1 & & & \\ 1 & 5 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 5 & 1 \\ & & & 1 & 6 \end{bmatrix}, \quad K_2 = \begin{bmatrix} 5 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 5 \end{bmatrix}$$

and K_0, K_1 and K_2 are $N \times N$ blocks. Thus, (45) is fully discretized as follows:

$$\begin{gathered} [A(\boldsymbol{u}_m^{\text{lex}}) + K^2]\boldsymbol{\delta u}^{\text{lex}} = -[(A(\boldsymbol{u}_m^{\text{lex}}) + K^2)\boldsymbol{u}_m^{\text{lex}} - K\tilde{\boldsymbol{u}}^{\text{lex}}] \\ \boldsymbol{u}_{m+1}^{\text{lex}} = \boldsymbol{u}_m^{\text{lex}} + \boldsymbol{\delta u}^{\text{lex}}, \quad m = 0, 1, \ldots, \quad \boldsymbol{u}_0^{\text{lex}} = \tilde{\boldsymbol{u}} \end{gathered} \tag{55}$$

On the other hand, the complications of formulating the matrix $A(\boldsymbol{u}) + K^2$ can be avoided by discretizing (45) in the manner of (24). For example the corresponding stencils for the approximation of $-\nabla \cdot [\phi'(|\nabla u|^2)\nabla v]$ are given at field cells by dividing the following by h^2:

(x_i, y_j) :

0	$-\phi'_{i,j+\frac{1}{2}}$	0
$-\phi'_{i-\frac{1}{2},j}$	$\phi'_{i\pm\frac{1}{2},j} + \phi'_{i,j\pm\frac{1}{2}}$	$-\phi'_{i+\frac{1}{2},j}$
0	$-\phi'_{i,j-\frac{1}{2}}$	0

(56)

and the stencils are given at the corner cells $\{(x_i, y_j) : 1 \le i, j, \le 2\}$ by dividing the following by h^2:

(x_1, y_2) :

0	$-\phi'_{1,2+\frac{1}{2}}$	0
0	$\phi'_{1+\frac{1}{2},2} + \phi'_{1,2\pm\frac{1}{2}}$	$-\phi'_{1+\frac{1}{2},2}$
0	$-\phi'_{1,2-\frac{1}{2}}$	0

(x_2, y_2) :

0	$-\phi'_{2,2+\frac{1}{2}}$	0
$-\phi'_{2-\frac{1}{2},2}$	$\phi'_{2\pm\frac{1}{2},2} + \phi'_{2,2\pm\frac{1}{2}}$	$-\phi'_{2+\frac{1}{2},2}$
0	$-\phi'_{2,2-\frac{1}{2}}$	0

(x_1, y_1) :

0	$-\phi'_{1,1+\frac{1}{2}}$	0
0	$\phi'_{1,1+\frac{1}{2}} + \phi'_{1+\frac{1}{2},1}$	$-\phi'_{1+\frac{1}{2},1}$
0	0	0

(x_2, y_1) :

0	$-\phi'_{2,1+\frac{1}{2}}$	0
$-\phi'_{2-\frac{1}{2},1}$	$\phi'_{2,1+\frac{1}{2}} + \phi'_{2\pm\frac{1}{2},1}$	$-\phi'_{2+\frac{1}{2},1}$
0	0	0

(57)

with stencils for other boundary cells given similarly. Note that the symmetry of the discretization can be seen from the identical weights in adjacent stencil entries, e.g., between (x_1, y_1) and (x_2, y_1), $\phi'_{1,1+1/2} = \phi'_{1,2-1/2}$. Now in the manner of (24) define the discretization of the second-order operator $-\nabla \cdot [\phi'(|\nabla u|^2)\nabla v]$ by $L_2(\boldsymbol{u})\boldsymbol{v}$ using the stencils shown above. Similarly, the stencils for the approximation of Kv are given at the corner cells $\{(x_i, y_j) : 1 \le i, j \le 2\}$ by

dividing the following by 8,

$$(x_1,y_2):\begin{array}{|r|r|r|}\hline 0&+1&0\\\hline 0&5&+1\\\hline 0&+1&0\\\hline\end{array}\qquad (x_2,y_2):\begin{array}{|r|r|r|}\hline 0&+1&0\\\hline +1&4&+1\\\hline 0&+1&0\\\hline\end{array}$$
$$(x_1,y_1):\begin{array}{|r|r|r|}\hline 0&+1&0\\\hline 0&6&+1\\\hline 0&0&0\\\hline\end{array}\qquad (x_2,y_1):\begin{array}{|r|r|r|}\hline 0&+1&0\\\hline +1&5&+1\\\hline 0&0&0\\\hline\end{array} \tag{58}$$

and the stencils for other cells are given similarly. Then in the manner of (24) define the discretization of the zeroth-order operator Kv by $L_0\boldsymbol{v}$ using the stencils shown above. So the discrete approximation to the full operator $-\nabla\cdot[\phi'(|\nabla u|^2)\nabla v]+K^2v$ is given by $L(\boldsymbol{u})\boldsymbol{v}=L_2(\boldsymbol{u})\boldsymbol{v}+L_0^2\boldsymbol{v}$ written as follows:

$$L(\boldsymbol{u})\boldsymbol{v}=\sum_{|k|,|l|\le 1}\boldsymbol{\sigma}_{k,l}(\boldsymbol{u})S_{k,l}\boldsymbol{v} \tag{59}$$

Thus, (45) is fully discretized as follows:

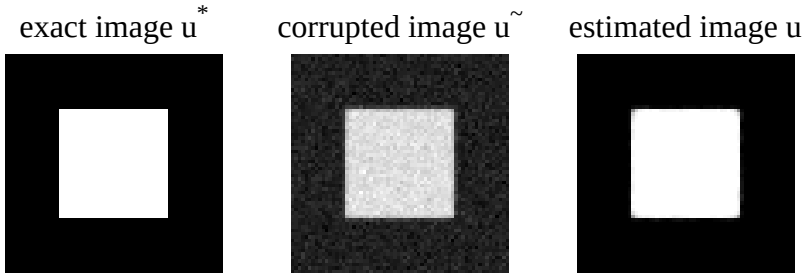
$$L(\boldsymbol{u}_m)\boldsymbol{\delta u}=-[L(\boldsymbol{u}_m)\boldsymbol{u}_m-L_0\tilde{\boldsymbol{u}}],\quad \boldsymbol{u}_{m+1}=\boldsymbol{u}_m+\boldsymbol{\delta u},\quad m=0,1,2,\ldots,\quad \boldsymbol{u}_0=\tilde{\boldsymbol{u}} \tag{60}$$

which can be solved in the manner of (33),

$$\begin{aligned}\boldsymbol{\delta u}_{n+1}(I_{\mathrm{R}}) &= \boldsymbol{\delta u}_n(I_{\mathrm{R}})+\frac{\omega}{\boldsymbol{\sigma}_{0,0}(I_{\mathrm{R}})}\left[\boldsymbol{f}(I_{\mathrm{R}})-\sum_{|k|+|l|\le 1}\boldsymbol{\sigma}_{k,l}(I_{\mathrm{R}})\boldsymbol{\delta u}(I_{\mathrm{R}}^{k,l})\right]\\ \boldsymbol{\delta u}_{n+1}(I_{\mathrm{B}}) &= \boldsymbol{\delta u}_n(I_{\mathrm{B}})+\frac{\omega}{\boldsymbol{\sigma}_{0,0}(I_{\mathrm{B}})}\left[\boldsymbol{f}(I_{\mathrm{B}})-\sum_{|k|+|l|\le 1}\boldsymbol{\sigma}_{k,l}(I_{\mathrm{B}})\boldsymbol{\delta u}(I_{\mathrm{B}}^{k,l})\right]\end{aligned} \tag{61}$$

with $\boldsymbol{f}=-[L(\boldsymbol{u}_n)\boldsymbol{u}_n-L_0\tilde{\boldsymbol{u}}]$.

An example of a solution to (41) is shown on the right in the following figure next to (a blurred and noisy image) $\tilde{u}$ shown in the middle and an ideal image u^* shown on the left.



This manner of formulating discretizations in terms of stencils is quite modular as can be seen from the following problem. Now u and v are given images which are not smooth, i.e., they possess edges where the intensity is discontinuous as seen in the figure below. On the other hand, the quotient of the two images is quite smooth. We want to estimate the smooth modulation s to satisfy $su\approx v$. This is done by minimizing:

$$J(s)=\int_\Omega[su-v]^2dx+\nu\sum_{|\alpha|=2}\left(\begin{array}{c}2!\\ \alpha!\end{array}\right)\int_\Omega|\partial^\alpha s|^2dx \tag{62}$$

where the sum over the multi-index α gives the Frobenius norm of the Hessian of s as the integrand. The optimality condition for the minimizer s is that it satisfy:

$$\left\{\begin{array}{rcll} \nu\Delta^2 s + u^2 s &=& uv, & \Omega \\ s_{nnn} = s_{nn} = s_{n\tau} &=& 0, & \partial\Omega \end{array}\right. \tag{63}$$

where s_n is the normal derivative of s and s_τ is the tangential derivative of s at $\partial\Omega$. The discretization of the biharmonic operator Δ^2 is given for the corner cells $\{(x_i, y_j) : 1 \le i, j \le 3\}$ by dividing the following by $2800h^4$,

$$\begin{array}{ccccc|ccccc|ccccc}
0 & 0 & -152 & 424 & 208 & 0 & 424 & 920 & 848 & 208 & 208 & 848 & 768 & 848 & 208 \\
0 & 0 & -592 & -2176 & 848 & 0 & -2176 & -3920 & -4352 & 848 & 848 & -4352 & -4512 & -4352 & 848 \\
0 & 0 & 4368 & -2256 & 768 & 0 & -2256 & 20400 & -4512 & 768 & 768 & -4512 & 24768 & -4512 & 768 \\
0 & 0 & -592 & -2176 & 848 & 0 & -2176 & -3920 & -4352 & 848 & 848 & -4352 & -4512 & -4352 & 848 \\
0 & 0 & -152 & 424 & 208 & 0 & 424 & 920 & 848 & 208 & 208 & 848 & 768 & 848 & 208 \\
\\
0 & 0 & -152 & 424 & 208 & 0 & 424 & 920 & 848 & 208 & 208 & 848 & 768 & 848 & 208 \\
0 & 0 & -592 & -2176 & 848 & 0 & -2176 & -3920 & -4352 & 848 & 848 & -4352 & -4512 & -4352 & 848 \\
0 & 0 & 3440 & -1960 & 920 & 0 & -1960 & 16960 & -3920 & 920 & 920 & -3920 & 20400 & -3920 & 920 \\
0 & 0 & -296 & -1088 & 424 & 0 & -1088 & -1960 & -2176 & 424 & 424 & -2176 & -2256 & -2176 & 424 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\\
0 & 0 & -152 & 424 & 208 & 0 & 424 & 920 & 848 & 208 & 208 & 848 & 768 & 848 & 208 \\
0 & 0 & -296 & -1088 & 424 & 0 & -1088 & -1960 & -2176 & 424 & 424 & -2176 & -2256 & -2176 & 424 \\
0 & 0 & 928 & -296 & -152 & 0 & -296 & 3440 & -592 & -152 & -152 & -592 & 4368 & -592 & -152 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array} \tag{64}$$

Let the corresponding discretization of $\nu\Delta^2$ with the natural boundary conditions by denoted by L_4. The discretization of the zeroth-order operator L_0, given by multiplication by u^2, is obtained by a single stencil weight of $u_{i,j}^2$ at the center of each stencil. So the discrete approximation to the full operator $\nu\Delta^2 + \boldsymbol{u}^2$ is given by $L = L_4 + L_0$ which can be written in the form (24). Thus, the discrete solution $\boldsymbol{s}$ to (63) satisfies:

$$L\boldsymbol{s} = \boldsymbol{u}\boldsymbol{v} \tag{65}$$

To solve this linear system iteratively requires to modify an iteration such as (33) so that it involves *nine* separate colors, since this is the minimum number such that all cells of one color may be updated simultaneously in terms of cells of other colors:

$$\mathrm{c} = 1, \ldots, 9$$
$$\boldsymbol{u}_{n+1}(I_\mathrm{c}) = \boldsymbol{u}_n(I_\mathrm{c}) + \frac{\omega}{\boldsymbol{\sigma}_{0,0}(I_\mathrm{c})}\left[\boldsymbol{u}(I_\mathrm{c})\boldsymbol{v}(I_\mathrm{c}) - \sum_{|k|+|l|\le 3} \boldsymbol{\sigma}_{k,l}(I_\mathrm{c})\boldsymbol{u}(I_\mathrm{c}^{k,l})\right] \tag{66}$$

Specifically, all nine cells $\{(x_i, y_j) : 1 \le i, j \le 3\}$ would have distinct colors, and then every cell at three horizontal or vertical steps away from a given cell would have the same color.

An example of a solution to (63) is shown on the right in the following figure next to the given images u and v shown respectively on the left and in the middle.