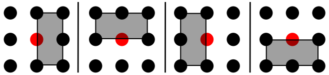
Kernels for Curvature Filter



For the most left case in above figure, we have following kernels

Filter	Regularization	Prior Surface	Kernel
GC	$\int \lvert\kappa_1\kappa_2\rvert \mathrm{d}\vec{x}$	Developable	$\begin{matrix} 0 & \frac{1}{2} & 0 \\ 0 & -1 & 0 \\ 0 & \frac{1}{2} & 0 \end{matrix}$
MC	$\int \lvert\kappa_1+\kappa_2\rvert \mathrm{d}\vec{x}$	Minimal Surface	$\begin{matrix} 0 & \frac{5}{16} & \frac{-1}{8} \\ 0 & -1 & \frac{5}{8} \\ 0 & \frac{5}{16} & \frac{-1}{8} \end{matrix}$
Bernstein	$\int \lvert\kappa_1+\kappa_2\rvert \mathrm{d}\vec{x}$	Minimal Surface	$\begin{matrix} 0 & \frac{1}{2} & 0 \\ 0 & -1 & 0 \\ 0 & \frac{1}{2} & 0 \end{matrix}$
TV	$\int \lvert\nabla U\rvert \mathrm{d}\vec{x}$	locally constant	$\begin{matrix} 0 & \frac{1}{5} & \frac{1}{5} \\ 0 & -1 & \frac{1}{5} \\ 0 & \frac{1}{5} & \frac{1}{5} \end{matrix}$

One example: derive the kernel for Bernstein filter

The five points in the left case are $(1,1,U_1)$,$(1,0,U_2)$,$(1,-1,U_3)$, $(0,-1,U_4)$ and $(0,1,U_5)$. According to **Bernstein Theorem**, these five points form a plane: $U(\hat{x},\hat{y}) = C_2\hat{x} + C_1\hat{y} + C_0$, where $\hat{x}$ and $\hat{y}$ denote the local coordinate. The parameters C_2, C_1 and C_0 can be found by minimizing the following quadratic form

$$T(C_2,C_1,C_0) = \sum_{i=1}^{5}(C_2\hat{x}_i + C_1\hat{y}_i + C_0 - U_i)^2 \,. \tag{1}$$

Letting $\frac{\partial T(C_2,C_1,C_0)}{\partial C_i} = 0$, we have

$$\begin{bmatrix} \sum \hat{x}_i^2 & \sum \hat{x}_i\hat{y}_i & \sum \hat{x}_i \\ \sum \hat{x}_i\hat{y}_i & \sum \hat{y}_i^2 & \sum \hat{y}_i \\ \sum \hat{x}_i & \sum \hat{y}_i & \sum 1 \end{bmatrix} \begin{bmatrix} C_2 \\ C_1 \\ C_0 \end{bmatrix} = \begin{bmatrix} \sum \hat{x}_i U_i \\ \sum \hat{y}_i U_i \\ \sum U_i \end{bmatrix} \,, \tag{2}$$

which is

$$\begin{bmatrix} 3 & 0 & 3 \\ 0 & 4 & 0 \\ 3 & 0 & 5 \end{bmatrix} \begin{bmatrix} C_2 \\ C_1 \\ C_0 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{3} U_i \\ U_1 + U_5 - U_3 - U_4 \\ \sum_{i=1}^{5} U_i \end{bmatrix} \,. \tag{3}$$

One example: derive the kernel for Bernstein filter

Since the red dot is the origin $(0, 0)$, the desired U is equal to $C_0 = \frac{U_4+U_5}{2}$ and $d_k = U - U_0 = \frac{U_4+U_5}{2} - U_0$, where U_0 is current pixel value.

Once we computed all $\{d_k\}$, we choose d_m that has minimal absolute value among $\{d_k\}$ to update U. This is Bernstein Filter.

Other kernels can be found in **my PhD thesis**. For example, the MC filter kernel is from Equation 6.12 in my thesis (only half window used). It seems that half window regression plays an essential role in preserving edges.