Masks are patterns used to define the weights used in averaging the neighbors of a pixel to compute some result at that pixel.
Expressing linear operations on neighborhoods

- many operations defined by cross correlation
- smoothing of image
- edge detection
  - ad hoc first derivative masks
  - ad hoc second derivative masks
  - LOG and DOG masks
- directional derivative masks
- special basis vector expansion of neighborhoods
  - look for energy matching certain patterns
  - Frei-Chen basis (const, edge, ripple, line, Laplacian)
  - Fourier (sine waves), Gabor wavelets
  - Laws texture masks (Ch 7)
Images as functions
Neighborhood operations

- Average neighborhood to remove noise or high frequency patterns
- Detect boundaries at points of contrast using gradient computation
- Can use median filtering to smooth while keeping boundaries sharp
Image processing examples

Histogram equalization; gamma correction; median filtering
Histogram equalization

Left image does not use all available gray levels. Image is recoded so that all gray levels are used and such that each gray level occurs in roughly the same number of pixels of the recoded image. (See algorithm in text, xv.)
Histogram equalization can darken a bright image, perhaps improving contrast

\[
cdf_x(i) = \sum_{j=0}^{i} p_x(j)
\]

\[
y = T(x) = cdf_x(x)
\]
Histogram equalization

\[
\begin{bmatrix}
52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\
63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\
62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\
63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\
67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\
79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\
85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\
87 & 79 & 69 & 68 & 65 & 76 & 78 & 94
\end{bmatrix}
\begin{bmatrix}
0 & 12 & 53 & 93 & 146 & 53 & 73 & 166 \\
65 & 32 & 12 & 215 & 235 & 202 & 130 & 158 \\
57 & 32 & 117 & 239 & 251 & 227 & 93 & 166 \\
65 & 20 & 154 & 243 & 255 & 231 & 146 & 130 \\
97 & 53 & 117 & 227 & 247 & 210 & 117 & 146 \\
190 & 85 & 36 & 146 & 178 & 117 & 20 & 170 \\
202 & 154 & 73 & 32 & 12 & 53 & 85 & 194 \\
206 & 190 & 130 & 117 & 85 & 174 & 182 & 219
\end{bmatrix}
\]
Histogram equalization

<table>
<thead>
<tr>
<th>Value</th>
<th>Count</th>
<th>Value</th>
<th>Count</th>
<th>Value</th>
<th>Count</th>
<th>Value</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>52</td>
<td>1</td>
<td>64</td>
<td>2</td>
<td>72</td>
<td>1</td>
<td>85</td>
<td>2</td>
</tr>
<tr>
<td>55</td>
<td>3</td>
<td>65</td>
<td>3</td>
<td>73</td>
<td>2</td>
<td>87</td>
<td>1</td>
</tr>
<tr>
<td>58</td>
<td>2</td>
<td>66</td>
<td>2</td>
<td>75</td>
<td>1</td>
<td>88</td>
<td>1</td>
</tr>
<tr>
<td>59</td>
<td>3</td>
<td>67</td>
<td>1</td>
<td>76</td>
<td>1</td>
<td>90</td>
<td>1</td>
</tr>
<tr>
<td>60</td>
<td>1</td>
<td>68</td>
<td>5</td>
<td>77</td>
<td>1</td>
<td>94</td>
<td>1</td>
</tr>
<tr>
<td>61</td>
<td>4</td>
<td>69</td>
<td>3</td>
<td>78</td>
<td>1</td>
<td>104</td>
<td>2</td>
</tr>
<tr>
<td>62</td>
<td>1</td>
<td>70</td>
<td>4</td>
<td>79</td>
<td>2</td>
<td>106</td>
<td>1</td>
</tr>
<tr>
<td>63</td>
<td>2</td>
<td>71</td>
<td>2</td>
<td>83</td>
<td>1</td>
<td>109</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ h(v) = \text{round} \left( \frac{\text{cdf}(v) - \text{cdf}_{\text{min}}}{(M \times N) - \text{cdf}_{\text{min}}} \times (L - 1) \right) \]

\[ h(78) = \text{round} \left( \frac{46 - 1}{63} \times 255 \right) = \text{round} (0.714286 \times 255) = 182 \]

MSU CSE 803
Histogram equalization

http://en.wikipedia.org/wiki/Histogram_equalization
Can define mapping of input gray level to output level \((xv)\)

Gamma correction: boost all gray levels

Boost low levels and reduce high
Smoothing an image by averaging neighbors (boxcar)
Properties of smoothing masks

- Coordinates of smoothing masks are positive and sum to one so that output on constant regions is the same as the input.
- The amount of smoothing and noise reduction is proportional to the mask size.
- Step edges are blurred in proportion to the mask size.

\[
\text{boxcar : } \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{Gaussian : } \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}
\]

Gaussian has better properties than the boxcar.
Types of ideal edges (in 1D)

These types are also present in 2D and 3D images and are complicated by orientation variations.
Boxcar smoothing filter example

box smoothing mask $M = [1/3, 1/3, 1/3]$

<table>
<thead>
<tr>
<th>$S_1$</th>
<th>12 12 12 12 12 12 24 24 24 24 24</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1 \otimes M$</td>
<td>12 12 12 12 12 16 20 24 24 24 24</td>
</tr>
</tbody>
</table>

(a) $S_1$ is an upward step edge

<table>
<thead>
<tr>
<th>$S_4$</th>
<th>12 12 12 12 12 24 12 12 12 12 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_4 \otimes M$</td>
<td>12 12 12 16 16 16 12 12 12 12 12</td>
</tr>
</tbody>
</table>

(d) $S_4$ is a bright impulse or “line”

So, reducing noise will also degrade the signal.
Linear smoothing smooths noise and blurs signal

Blur: step is now ramp

Input image

Row after 5x5 mean filter

MSU CSE 803
Gaussian smoothing

Gaussian smoothing mask $M = [1/4, 1/2, 1/4]$

<table>
<thead>
<tr>
<th>$S_1$</th>
<th>12</th>
<th>12</th>
<th>12</th>
<th>12</th>
<th>12</th>
<th>24</th>
<th>24</th>
<th>24</th>
<th>24</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1 \otimes M$</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>15</td>
<td>21</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
</tbody>
</table>

(a) $S_1$ is an upward step edge

<table>
<thead>
<tr>
<th>$S_4$</th>
<th>12</th>
<th>12</th>
<th>12</th>
<th>12</th>
<th>24</th>
<th>12</th>
<th>12</th>
<th>12</th>
<th>12</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_4 \otimes M$</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>15</td>
<td>18</td>
<td>15</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>

(d) $S_4$ is a bright impulse or “line”
Median filter replaces center with neighborhood median, not mean

Noisy row of checkers image

Mean filtering smooths signal and ramps the boundary

Median filter smooths signal and preserves sharp boundary
Median filter is not linear

- Algorithm requires comparisons and is more expensive than using mask
- Can sort all $\text{NxN}$ pixel values and pick middle
- Do not need totally sorted data: $O(N)$ algorithm exists

$$X(n) = aX(n1) + bX(n2) \rightarrow Y(n) = aY(n1) + bY(n2)$$
Scratches removed by using a median filter

Thin artifact removed, sharp boundaries preserved.
Finding boundary pixels

Computing derivatives or gradients to locate region change.
2 rows of intensity vs difference
Differencing used to estimate 1st and 2nd derivatives

First differences
2nd differences

Masks represent the first and 2nd differences

MSU CSE 803
Step edges X mask $[-1, 0, +1]$

Step edge is detected well, but edge location imprecise.

(a) $S_1$ is an upward step edge

(b) $S_2$ is a downward step edge
Ramp and impulse X mask [-1, 0, +1]

(c) $S_3$ is an upward ramp

(d) $S_4$ is a bright impulse or “line”

Ramp edge now yields a broad weak response. Impulse response is a “whip”, first up and then down.
2nd derivative using mask \([-1, 2, -1]\)

Response is zero on constant region and a “double whip” amplifies and locates the step edge.
2nd derivative using mask [-1, 2, -1]

(c) $S_3$ is an upward ramp

(d) $S_4$ is a bright impulse or “line”

Weak response brackets the ramp edge. Bright impulse yields a double whip with gain of 3X original contrast.
Estimating 2D image gradient

Can estimate the column (x) gradient across 3 columns

$$\frac{\partial f}{\partial x} \equiv f_x \approx \frac{1}{3} \left[ \frac{(I[x+1,y] - I[x-1,y])}{2} + \frac{(I[x+1,y-1] - I[x-1,y-1])}{2} + \frac{(I[x+1,y+1] - I[x-1,y+1])}{2} \right]$$

Can estimate the row (y) gradient across 3 rows

$$\frac{\partial f}{\partial y} \equiv f_y \approx \frac{1}{3} \left[ \frac{(I[x,y+1] - I[x,y-1])}{2} + \frac{(I[x-1,y+1] - I[x-1,y-1])}{2} + \frac{(I[x+1,y+1] - I[x+1,y-1])}{2} \right]$$
Gradient from 3x3 neighborhood

Estimate both magnitude and direction of the edge.

\[ f_y = \frac{1}{3} \left( \frac{(38-12)}{2} + \frac{(66-15)}{2} + \frac{(65-42)}{2} \right) = \frac{(13 + 25 + 11)}{3} = 16 \]

\[ f_x = \frac{1}{3} \left( \frac{(65-38)}{2} + \frac{(64-14)}{2} + \frac{(42-12)}{2} \right) = \frac{(13 + 25 + 15)}{3} = 18 \]

\[ \theta = \tan^{-1} \left( \frac{16}{18} \right) = 0.727 \text{ rad} \]
\[ = 42 \text{ degrees} \]

\[ |\nabla f| = \left( 16^2 + 18^2 \right)^{1/2} = 24 \]
Alternative masks for gradient

<table>
<thead>
<tr>
<th>Mask</th>
<th>$M_x$</th>
<th>$M_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prewitt</td>
<td>$\begin{bmatrix} -1 &amp; 0 &amp; 1 \ -1 &amp; 0 &amp; 1 \ -1 &amp; 0 &amp; 1 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 1 &amp; 1 &amp; 1 \ 0 &amp; 0 &amp; 0 \ -1 &amp; -1 &amp; -1 \end{bmatrix}$</td>
</tr>
<tr>
<td>Sobel</td>
<td>$\begin{bmatrix} -1 &amp; 0 &amp; 1 \ -2 &amp; 0 &amp; 2 \ -1 &amp; 0 &amp; 1 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 1 &amp; 2 &amp; 1 \ 0 &amp; 0 &amp; 0 \ -1 &amp; -2 &amp; -1 \end{bmatrix}$</td>
</tr>
<tr>
<td>Roberts</td>
<td>$\begin{bmatrix} 0 &amp; 1 \ -1 &amp; 0 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 1 &amp; 0 \ 0 &amp; -1 \end{bmatrix}$</td>
</tr>
</tbody>
</table>

- Prewitt: gradient of least-squares plane through the 9 pixels
- Sobel: scale of 1/8 easy to compute by shifting
- Roberts: faster because of $2 \times 2$ neighborhood
- Roberts: gradient computed across diagonals
Prewitt versus Sobel masks

\[
\text{Prewitt Masks:} \quad \frac{\partial f}{\partial x} \approx \frac{1}{6}(M_x \circ N_8[x, y]) \\
\frac{\partial f}{\partial y} \approx \frac{1}{6}(M_y \circ N_8[x, y])
\]

\[
\text{Magnitude:} \quad | \nabla f | \approx \sqrt{\frac{\partial f^2}{\partial x} + \frac{\partial f^2}{\partial y}}
\]

\[
\text{Direction:} \quad \theta \approx \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)
\]

Sobel mask uses weights of 1,2,1 and -1,-2,-1 in order to give more weight to center estimate. The scaling factor is thus 1/8 and not 1/6.
Computational short cuts

- for magnitude use $\max(\left| \frac{\partial f}{\partial y} \right|, \left| \frac{\partial f}{\partial x} \right|)$

- for magnitude use $\left| \frac{\partial f}{\partial y} \right| + \left| \frac{\partial f}{\partial x} \right|$

- omit direction computation
Properties of derivative masks

- Coordinates of derivative masks have opposite signs in order to obtain a high response in signal regions of high contrast.
- The sum of coordinates of derivative masks is zero so that a zero response is obtained on constant regions.
- First derivative masks produce high absolute values at points of high contrast.
- Second derivative masks produce zero-crossings at points of high contrast.
Interpret the properties of masks using the theory of vectors and dot products.