Linear Filters

Noise:
Salt & Pepper Noise: Random occurrences of black and white pixels
Impulse Noise: Random occurrences of white pixels
Gaussian Noise: Variations in intensity drawn from a Gaussian normal distribution
Filters:
Linear: Obeys Superposition, Shift-Invariant, Casual, and Stable
Non-Linear: Opposite of linear
Smoothing Filter: Values positive, $\sum F = 1$, amount of smoothing proportional to mask size, remove 'high-frequency'components; 'low-pass'filter
Derivative Filter: Opposite signs to get high response in areas of high contrast, $\sum F = 0$, High absolute value when high contrast
Correlation Filter(Cross-Correlation): Pixel is linear combination of surrounding pixels, G=H$\otimes$F.
Gaussian Filter: Linear, smoothing, σ=variance, kernel=size of mask
Sharpening Filter: Accentuates differences with local average, subtraction
Convolution: Linear, Flip the filter in both dimensions (bottom to top, right to left), then apply cross-correlation, G=H$\star$F
Shift Invariant: Operator behaves the same everywhere; the value of the output depends on the pattern in the image neighborhood, not the position of the neighborhood
Superposition: The response to a sum of inputs is the sum of the responses to the individual inputs
Seperability: A 2D filter is separable if it can be written as the outer product of two 1D filters
Median Filter: Non-linear, no new pixel values, removes spikes, good for impulse + salt & pepper noise, edge preserving
Laplacian Filter: Hybrid images, Unit impulse - gaussian $\approx$ laplacian of gaussian

Edge Detection

Edge: rapid change in image intensity, extrema of the first derivative, zero-crossings of the second derivative
Steps: 1. Smooth/Suppress noise, 2. Edge enhancement/Filter for contrast, 3. Edge localization /Local maxima/Threshold/Thinning
Canny Edge Detector:
Algorithm: 1. Filter image with derivative of Gaussian then get magnitude/direction of gradient, 2. Non-maximum suppression, 3. Linking and thresholding (hysteresis)
Property: Threshold strong edges and weak edges using two thresholds, then keep weak edges only if connect to strong edges.
Property: Filtering a signal f with a Gaussian and then calculating its gradient is the same as filtering the signal f with the first order derivative of the Gaussian.

Seam Carving

Energy: $\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$
Seam: Path of least energy, 8-connected, 1 pixel wide
Cumulative Energy Map: M(i,j)=Energy(i,j)+min(M(i-1,j-1),M(i-1,j),M(i-1,j+1))

Template Matching

Template: Uses filters that match effect intended to search
Algorithm: Use normalized(Controls relative brightness) cross-correlation score to find template in image

Binary Image Analysis

Steps: 1. Threshold image into binary form, 2. Clean using morphological operations, 3. Extract separate blobs/Connected components, 4. Describe blobs with region properties
Otsu's Method: Find threshold that minimizes intra-class variance
Morphological Operations:
Dilation: Closes gaps and fills holes, I$\oplus[1]_{i\times i}$
Erosion: Erode connected components, Shrink features, Removes bridges/branches/noise, I$\ominus[1]_{i\times i}$
Opening: Erosion then dilation, removes noise/small objects, keeps original shape
Closing: Dilation then erosion, closes small holes inside

or small black points on the object.

Texture

Segmentation: Analyze/represent texture, group image regions with consistent texture
Synthesis: Generate new texture patches/images
Filter Banks: Collection of (d) features, d-dimensional vector, contain combination of scales, orientation, and patterns
First-order Markov Random Field: Each pixel depends on its neighbors, P(X|A,B,C,D)