The Digital Image

Problems: Transmission interference, compression artifacts, spilling, scratches, sensor noise, bad contrast and resolution, motion blur
Pixel: Discrete samples of an continuous image function.

Charge Coupled Device (CCD)

Has an array of photosites (a bucket of electrical charge) that charge proportional to the incident light intensity during exposure. ADC happens line by line.
Bloooming: oversaturation of finite capacity photosites causes the vertical channels to "flood" (bright vertical line)
Bleeding/Smearing: While shifting down, the pixels above get some photons on bright spot with electronic shutters.
Dark Current: CCDs produce thermally generated charge they give non-zero output even in darkness (fluctuates randomly) due to spontaneous generation of electrons due to heat $\rightarrow$ cooling.
can be avoided by cooling, worse with age.

CMOS:

Same sensor elements as CCD, but each sensor has its own amplifier $\rightarrow$ faster readout, less power consumption, cheaper, more noise.
more noise, lower sensitivity
vs CCD cheaper, lower power, less sensitive, per pixel amplification random pixel access, no blooming, on chip integration

Sampling methods

Cartesian (grid), hexagonal, non-uniform
Quantization: Real valued function will get digital values (integers). A lossy process (original cannot be reconstructed). Simple version: equally spaced $2^b = \#bits$ levels
Bilinear Interpolation: TODO
Resolution: Image resolution (cropping), geometric resolution (#pixels per area), radiometric resolution (#bits per pixel, color)
Image noise: commonly modeled by additive Gaussian noise: $I(x,y) = f(x,y) + c$, poisson noise (shot noise for low light, depends on signal & apature time), multiplicative noise: $I = f + f * c$, quantization errors, salt-and-pepper noise SNR or peak SNR is used as an index of image quality $c \sim N(0,\sigma^2), p(c) = \frac{1}{\sigma\sqrt{2\pi}} * e^{\frac{(c-\mu)^2}{2*\sigma^2}}, SNR: S = \frac{F}{\sigma}$ where $F = \frac{1}{XY}\sum_{x=1}^{x}\sum_{y=1}^{y} f(x,y)$

Color cameras:

Prism need 3 sensors and good alignment
Filter mosaic coat directly on sensor
Wheel multiple filters in front of same sensor
New CMOS sensor layers that absorb color at different depths $\rightarrow$ better quality

Image Segmentation

Complete segmentation

Finite set of non-overlapping regions that cover the whole image $I = \bigcup_{i=1}^{n} R_i$ and $R_i \cap R_j = \emptyset \; \forall i,j, i \neq j$
Thresholding: simple segmentation by comparing greylevel with a threshold to decide if in or out.
Chromakeying: when planning to segment, use special backgroundcolor. (Problems variations due to lighting, noise, ... mixed pixels (hard α-mask does not work)) $I_\alpha = |I - g| > T$

Receiver Operating Characteristic (ROC) analysis:

ROC curve characterizes performance of binary classifier Classification errors: False negative (FN), false positives (FP)
ROC curve plots TP fraction $\frac{TP}{TP+FN}$ vs FP fraction $\frac{FP}{FP+TN}$
TODO

Picxel connectivity

TODO
Connected component raster scanning: scanning row by row, if foreground & label if connected to other label, alse give new label. (second pass to find equivalent labels)
Improve: when region found, follow border, then carry on (contour-based method)

Region growing

Start with seed point or region, add neighboring pixels that satisfy a criteria defining a region until we include no more pixels.
Seed region: by hand or automatically by conservative Thresholding
Inclusion criteria: greylevel thresholding, greylevel distribution model (include if $(I(x,y) - \mu^2)^2 < (n\sigma)^2$ and update μ and σ after each iteration) color or texture information
Snakes: active contour, a polygon and each point moves away from seed while criteria is met (can have smoothness constraint) Iteratively minimize enery function $E = E_{tension} + E_{stiffness} + E_{image}$

background subtraction

simple: $I_\alpha = |I - I_{bg}| < T$ better: $I_\alpha = \sqrt{(I - I_{bg})^T \Sigma^{-1} (I - I_{bg})}$ where Σ is the background pixel appearance covariance matrix, computed seperately for each pixel.

Morphological operators

Logical transformations based on comparison of neighboring pixels
erode delete FG pixels with 8-connected BG pixels
dilate every BG pixels with 8-connected FG pixel make a FG pixel
Uses: smooth regions, remove noise and artifacts.

Image Filtering

Modify the pixels of an image based on some function of the local neighborhood of the pixels- If sum greater 1 get brighter, if smaller darker.
separable: if a kernel can be written as a product of two simpler filters $\rightarrow$ computationally faster (filter $P \times Q$, image $N \times M: (P+Q) * NM$ instead of $PQNM$)
shift invariant: Doing the same thing, applying the same function over all pixels (in the formula below if K does not depend on x, y)
linear: linear combination of neighbors can be written as: $I'(x,y) = \sum_{(i,j)\in\mathbb{N}(i,j)} K(x,y,i,j) I(x+i, y+j)$ TODO
Filter at edges: clip filter (black), wrap around, copy edge, reflect across edge, vary filter near edge

Correlation

$I' = K \circ I, I'(x,y) = \sum_{(i,j)\in\mathbb{N}(i,j)} K(i,j) I(x+i, y+j)$
e.g. template matching: search for best match by minimizing mean squared error or maximizing area correlation. (remove mean (from filter, from image) to avoid bias)

Convolution

$I' = K * I, I'(x,y) = \sum_{(i,j)\in\mathbb{N}(i,j)} K(i,j) I(x-i, y-j)$
if $K(i,j) = K(-i,-j) \implies corrolation = convolution$
Continuous: $(f*g)(t)$
$= \int_{-\infty}^{\infty} f(\tilde{t}) g(t-\tilde{t}) d\tilde{t}$
$= \int_{-\infty}^{\infty} f(t-\tilde{t}) g(\tilde{t}) dt$

Kernels

Box filter: all same values normalized to sum = 1
Gaussian Kernel: $K(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$ is separable, e.g. $\sigma = 1$
Rotationally symmetric, has single lobe, single lobe in frequency domain, simple relationship to σ easy to implement efficiently, neighbors decrease monotonically, no corruption from higher frequency.
Subtracting one from central element of low-pass filter gives a high-pass filter with inverted sign, because.
$(f - \delta) * a = f * a - \delta * a = f * a - a = -(a - (f * a))$
Band pass filter: do LPF and HPF with cutoffs $D_{LP} < D_{HP}$
Band reject filter: do LPF and HPF with cutoffs $D_{LP} > D_{HP}$
Image sharpening: increases high frequency components to enhance edges: $I' = I + \alpha |K * I|$ K: high-pass filter, α: scalar $\in [0,1]$

Features

Edge Detection

How to tell if there is an edge? Local maxima of the first derivative and the zero crossing of the second derivative.
Edge detection filters:
Sobel:
$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, K_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Prewitt:
$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, K_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
Roberts:
$K_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, K_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$

Gradient Magnitude:
$M(x,y) = \sqrt{(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2}$
Gradient Angle:
$\alpha(x,y) = \tan^{-1}(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x})$

Laplacian operator

detect discontinuities by considering second derivative

TODO $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ or $\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ are discrete space approximations. Is isotropic(rotationally invariant), zero crossings make edge locations. Sensitive to fine details and noise ($\rightarrow$ smoothing before applying).
blur image first (LoG)
Laplacian of Gaussian (LoG): convolve gaussian blurring and laplacian operator in LoG operator (cheaper) $LoG(x,y) = -\frac{1}{\pi\sigma^4}(1-\frac{x^2+y^2}{2\sigma^2})e^{-\frac{x^2+y^2}{2\sigma^2}}$

Canny Edge Detector: 5 Steps

1. smooth image with a Gaussian filter
2. compute gradient magnitude and angle
3. apply non-maximum suppression to gradient magnitude image (Quantize edge normal to one of four directions: horizontal, +45°, vertical, -45°. If $M(x,y)$ smaller thn either of its neighbors in edge normal direction suppress, else keep)
4. Double thresholding to detect strong and weak edge pixels
5. Reject weak edge pixels not connected to strong edge pixels

Hough Transform

Fitting a straight line to a set of edge TODO
Fr circles: if r known: calculate circles with radius r around edge pixels $\rightarrow$ intersection of circles gives center.
where lots of them meet is the center of a circle. else: use 3D hough transform with parameters (x_0, y_0, r)

Corner Detection

Edges are only well localized in one direction $\rightarrow$ detect corners.
Desirable properties: Accute localization, invariance against shift, rotation, scale, brightness change, robust against noise, high repeatability
Linear approximation for small $\Delta x \Delta y$: (Taylor) $f(x+\Delta x, y+\Delta y) \approx f(x,y) + f_x(x,y)\Delta x + f_y(x,y)\Delta y$

Local displacement sensitivity (Harris corners)

$S(\Delta x, \Delta y) = (\Delta x \Delta y)(\sum_{x,y \in window}) \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$
$\approx SSD$
Find points where $\min \Delta^T M \Delta$ is large for $||\Delta|| = 1$ i. e. maximize the eigenvalues of M
$C(c,y) = \det(M) - k * trace(M)^2 = \lambda_1 * \lambda_2 + k * (\lambda_1 + \lambda_2)$ Harris cornerness: Measure of cornerness
Robustness of Harris corner detector: Invariant to brightness offset, invariant to shift and rotation but not to scaling! $\lambda_1 >> \lambda_2 \rightarrow$ edge, λ_1 and λ_2 large $\rightarrow$ corner, else $\rightarrow$ flat region.
not scale invariant: TODO
Overcome issues: look for strong DoG response or consider local maxima in position and scale space, Gaussian weighing.

Lowe's SIFT features

Look for strong responses of difference of Gaussians (DoG) filter, only look at local maxima in both position and scale.

DoG: $DoG(x,y) = \frac{1}{k} * e^{-\frac{x^2+y^2}{(k\sigma)^2}} - e^{-\frac{x^2+y^2}{\sigma^2}}$ e.g. $k = \sqrt{2}$
Orientation: create histogram of local gradient directions computed at selected scale, assign canonical orientation at peak of smoothed histogram. Get a SIFT descriptor (threshold image gradients are sampled over 16×16 array of locations in scale space) and do matching with these. Invariant to scale, rotation, illumination and viewpoint.

Fourier Transform

Aliasing: Happens when undersampling e.g. taking every second pixel, else characteristic errors appear: typically small phenomena look bigger, fast phenomena look slower. (e.g. wagon wheels backwards in movies, checkerboards misrepresented)

Fourier Transform

Represent function on a new basis with basis elements $e^{i2\pi(ux+vy)} = \cos(2\pi(ux+vy)) + i\sin(2\pi(ux+vy))$
$F(u) = \int_{-\infty}^{\infty} f(x)e^{-i2\pi fx}dx$, **2D:** $F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-2\pi i(ux,vy)}dxdy$
For images: transformed image $\rightarrow F = U * f \leftarrow$ vectorized image, U: Fourier matrix
For discrete: $F(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y)e^{-2\pi i(\frac{ux}{N}, \frac{vy}{M})}$
1D-periodic function: $f(t) = \sum_{n=-\infty}^{\infty} c_n e^{\frac{i2\pi nt}{T}}$, $c_n = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)e^{\frac{-i2\pi nt}{T}}dt$

Properties of Fourier transform

Linearity: $F(ax(t) + by(t)) = aX(t) + bY(t)$
Time Shift: $F(x(t \pm t_0)) = X(t)e^{\pm i2\pi f t_0}$
Frequency Shift: $F(e^{i2\pi f_0 t}x(t)) = X(f - f_0)$
Scaling: $F(x(at)) = \frac{1}{|a|}X\left(\frac{f}{a}\right)$
Convolution: $F(x(t) * y(t)) = X(f) \cdot Y(f)$
Duality: $F(X(t)) \longleftrightarrow x(-f)$