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 → cooling.
can be avoided by cooling, worse with age.

CMOS:

Same sensor elements as CCD, but each sensor has its own amplifier → 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 & aperture time), multiplicative noise: $I = f + f \cdot c$, quantization errors, salt-and-pepper noise. SNR or peak SNR is used as an index of image quality $c \sim N(0, \sigma^2)$, $p(c) = \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp\left(-\frac{(c-\mu)^2}{2\sigma^2}\right)$, 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 → 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

Pixel 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 → 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 $\in [0,1]$ds

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 (→ 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 $\to$ 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 $\to$ 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)

$S(\Delta x,\Delta y) = (\Delta x \Delta y)\sum_{x,y\in \text{window}}\begin{bmatrix} f_x^2 & f_xf_y\\ f_xf_y & f_y^2\end{bmatrix}\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix} \approx$
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! $\lambda_1 >> \lambda_2 \to$ edge, λ_1 and λ_2 large $\to$ corner, else $\to$ 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 Gausians (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.

1 Fourier Transformation

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 $\to F = U*f \leftarrow$ 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: $F(X(t)) \longleftrightarrow x(-f)$
Sampling:
A sampling function $s(t)$ which is an impulse train with period T and its Fourier transform $S(f)$:
$S(t) = \sum_{n=-\infty}^{\infty}\delta(t-nT)$
$S(f) = \frac{1}{T}\sum_{n=-\infty}^{\infty}X(f-\frac{n}{T})$ where $\delta(*)$ Dirac-delta function
A continuous signal can be sampled by multiplying with $s(t)$: $x_s(t) = x(t)s(t)$To compute the Fourier Transform of $x_s(t)$, we can use the convolution theorem:

$F(x_s(t)) = X(t)*S(t) = \frac{1}{T}\sum_{n=-\infty}^{\infty}\delta(f-\frac{n}{T}) * X(t) = \frac{1}{T}\sum_{n=-\infty}^{\infty}X(f-\frac{n}{T})$

Sampling in 2D:$sample_{2D}(f(x,y)) = \sum_{i=\infty}^{\infty}\sum_{j=\infty}^{\infty} f(x,y) * \delta(x-i,x-j) = f(x,y)\sum_{i=\infty}^{\infty}\sum_{j=\infty}^{\infty}\delta(x-i,x-j)$
DFT:
$F(f(x,y))(\sum_{i=\infty}^{\infty}\sum_{j=\infty}^{\infty}\delta(x-i,x-j))F(f(x,y)) * F(\sum_{i=\infty}^{\infty}\sum_{j=\infty}^{\infty}\delta(x-i,x-j)) = \sum_{i=\infty}^{\infty}\sum_{j=\infty}^{\infty}F(u-i,v-j)$
Dirac Delta Function:
$\int_{-\infty}^{\infty}\delta(t)dt = 1$ and $\delta(t) = \begin{cases}0 & \text{for } x\neq 0\\ und. & \text{for } x = 0\end{cases}$
Sifting Property:
$\int_{-\infty}^{\infty} f(t)\cdot\delta(x-a)dx = f(a)$
Dirac Comb:
$\text{III}_T(x) = \sum_{n=-\infty}^{\infty}\delta(t-nT)$,
sampling = product with this
TODO!
Box Filter: $\text{rect}(t) = \begin{cases}1, & \text{if } |t|\leq\frac{1}{2}\\ 0, & \text{otherwise.}\end{cases}$
Triangle Filter: $\text{tri}(t) = \begin{cases}1-|t|, & \text{if } |t|\leq 1\\ 0, & \text{otherwise.}\end{cases}$

Fourier transform of important functions

TODO

Nyquist Sampling theorem

The sampling frequency must be at least twice the highest frequency $w_s \geq 2w$ If not the case: band limit before with low-pass filter. Perfect reconstruction: $sinc(x) = \frac{sin(\pi x)}{\pi x}$
Why should this hold? Function $f(t)$, sampling function $S_{\Delta t}(t)$ with sampling frequency w_s. Fourier transform of the sampled function can be derived as

$$\begin{aligned}\tilde F(u) &= F(f(t)\cdot S_{\Delta t}(t))\\ &= F(u)*S_{\Delta t}(w)\\ &= \int_{-\infty}^{\infty} F(\tilde t)S_{\Delta t}(w-\tilde t)d\tilde t\\ &= \int_{-\infty}^{\infty} F(\tilde t)\frac{1}{\Delta T}\sum_{n=-\infty}^{\infty}\delta(w-\tilde t-\frac{n}{\Delta T})d\tilde t\\ &= \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty}F(w-nw_s).\end{aligned}$$

If we want to reconstruct the signal $f(t)$ from F and $S_{\Delta t}$, $F(w)$ cannot overlap with its neighbors $F(w-w_s)$ and $F(w+w_s)$. Thus, w_s should be larger than w_n. Highest frequency of $f(t)$.

Image restoration problem: $f(x)\to h(x)\to g(x)\to\tilde h(x)\to f(x)$

The "inverse" kernel $\tilde h(x)$ should compensate $h(x)$. May be determined by: $F(\tilde h)(u,v)\cdot F(h(u,v)) = 1$
Problems: Convolution with kernel k may cancel out some frequencies & noise amplification.
Avoid: Regularization: $F(\tilde h)(u,v) = \frac{F(h)}{|F(h)|^2+\varepsilon}$ avoid singularities

2 Optical Flow

Apparent motion of brightness patterns use extracted feature points and commpute their velocity vectors projection of 3D velocity vectors on I
Problem: cannot distingish motion from changing lighting! also estimate observed projected motion field normal flow not always well defined
Key assumptions:
Brightness constancy: Proection of the same point looks the same in every frame.
Small motion: Points do not move far
Spatial coherence: Points move like their neighbors
Brightness constancy constraint: $I(x,y,t-1) = I(x+u,y+v,t)I = Intensity$
Small motion $\to$ can linearize with Taylor expansion:
$I(x,y,t-1)\approx I(x,y,t-1)+\frac{\partial I}{\partial x}u+\frac{\partial I}{\partial y}v+\frac{\partial I}{\partial t}\approx 0$ or shorthand $I_x\cdot u+I_y\cdot v+I_t\approx 0$
move $I-t$ on one side, vectorize unknowns. For LK, sum up over a window of pixes
Apature problem: 1 equation, 2 unknowns cannot determine exact location, take normal flow. TODO
Add additional smoothness constraint:
$e_s = \int\int(u_x^2+u_y^2+v_x^2+v_y^2)dxdy\ close\approx parallel$

Besides OF constraint:

$e_c = \int\int (I_x u + I_y v + I_t)^2 dxdy$ Minimize $e_s + \lambda e_c$

Lukas-Kanade

Assume same displacement for $N \times M$ window $\rightarrow$ linear least squares problem:

$$\begin{bmatrix} I_x(x_1,y_1) & I_y(x_1,y_1) \\ \vdots & \vdots \\ I_x(x_{NM},y_{NM}) & I_y(x_{NM},y_{NM}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(x_1,y_1) \\ \vdots \\ I_t(x_{NM},y_{NM}) \end{bmatrix} \implies$$

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_y I_x & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

When solvable? $A^T A$ invertible, eigenvalues λ_1, λ_2 large, $\frac{\lambda_1}{\lambda_2}$ small

Errors: motion is large(r than a pixel)

$\rightarrow$ iterative refinement and coarse-to-fine estimation.

A point does not move like its neighbors

$\rightarrow$ motion segmentation.

Brightness constancy does not hold:

$\rightarrow$ exhaustive neighborhood search with normalized corrolation.

KLT feature tracker: to find patches where LSE well-behaved $\rightarrow$ LK-flow

Iterative refinement: Estimate velocity, warp using estimate, refine,...

Coarse-toFine Estimation: Image Pyramid. Start small, compute OF, rescale, take larger and initialize with last estimate

Applications: Image stabilization (get flow between two frames and warp image using same OF for all pixels s.st. OF close to 0) frame interpolation, video compression, object tracking, motion segmentation **Affine motion:** $I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \approx 0$ TODO

SSD tracking: For large displacements: match template against each pixel in small area around, match measure can be (normalized) correlation or SSD choose max. as match (sub-pixel also possible)

Bayesan Optical Flow: Some low-level motion illusions can be explained by adding an underlying model to LK-tracking e.g. brightness constancy with noise.

3 Video Compression

Interlaced video format: 2 temporally shifted half images

$\rightarrow$ increase frequency, decrease spatial resolution $\rightarrow$ not progressive

Lossy video compression: take advantage of redundancy spatial correlation between pixels, temporal correlation between frames

$\rightarrow$ basically drop perceptually unimportant details

with optical flow: Encode optical flow based on previous frame can cause blocking artifacts, does not work well for lots of movemen, fast movement and scene changes.

If temporal redundancy fails $\rightarrow$ use motion-compensated prediction

Types of coded frames:

I-Frame: Intra-coded frame, coded independently of all others

P-Frame: Predictively coded frame, based on previously coded frame

B-Frame: Bi-directionally coded frame, based on previous & future

Block-Matching Motion Estimation:

Is a type of temporal redundancy reduction

Motion Estimation Algorithm ME

1. Partition frame into blocks (e.g. 16×16 pixels)
2. For each block, find the best matching block in reference frame

Metrics for best match: sum of differences or squared sum of diff.

Candidate blocks: All blocks in e.g. 32×32 pixel area

Search strategies: Full search, partial (fast) search

Motion Compensation Algorithm MC Use the best matching of reference frame as prediction of blocks in current frame

$\rightarrow$ gives motion vectors & MC prediciotn error or residual (encode with conventionl image coder)

Motion Vector: relative horizontal & vertical offsets of a given block from one frame to another

Not limited to integer-pixel offsets, can use half-pixel ME to capture sub-pixel motion.

Half-pixel ME (coarse-fine) algorithm:

1. Coarse step: find best integer move
2. Fine step: refine by spatial interpolation and best-matching

Advantages and disadvanages

+ good, robust performance, one MV per block $\rightarrow$ useful for compression, simple periodic structure (GoP)

- assumes translational motion (fails for complex motion)

$\rightarrow$ codes these frames/blocks without prediction produces blocking artifacts

MPEG-GoP IBBPBBPBBI dependencies between frames

Scalable Video Coding:

Decompose video into multiple layers of prioritized importance: e.g.

temporal scalability: Include B-frames or not

spatial scalability: Base resolution + upsampling difference

SNR scalability: Base with coarse quatizer + fine quantizer

Benefits: Adapting to different bandwidths, facilitates error resiliency by identifying more and less important bits.

4 Unitary Transforms

Vectorization: interpret image as vector row-by-ow: $I = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}$

linear image processing: can be written as $\vec{g} = H\vec{f}$

Image collection (IC): $F = [f_1, f_2 ... f_n]$

Autocorrelation matrix $Rff = \frac{F \cdot F^H}{N}$ its Eigenvector with largest Eigenvalue is direction of largest variance among pictures.

Unitary transform: for transform A iff $A^H = A^{-1}$

if real-valued $\rightarrow$ orthonormalevery unitary transform is a rotation + sign flip, length conserved

Energy conservation: $||\vec{C}||^2 = \vec{C}^H C = \vec{f}^H A^H A f = ||\vec{f}||^2$

Karhunen-Loeve Transform Same as PCA. Order by decreasing eigenvalues

Energy concentration property: no other unitary transform packs as much energy in the first J coefficients (for arbitrary J) and mean squared approximation error by choosing only first J coefficients is minimized.

Optimal energy concentratioin of KLT consider truncated coefficient vector $\vec{b} = I_J \vec{c}$ (I_J: identity matrix with first J columns) Energy in first J coefficients for an arbitrary transform A : $E = Tr(R_{bb}) = Tr(I_J R_{cc} I_J) = Tr(I_J A R_{ff} A^H I_J) = \sum_{k=0} J = 1 a_k^T R_{ff} a_k^*$ where a_k^T is $k-th$ row of A. Lagrangian cost function to enforce unit-length basis vectors: $L = E + \sum_{k=0}^{J-1} \lambda_k (1 - a_k^T a_k^*) = \sum_{k=0}^{J-1} a_k^T R_{ff} a_k^* + \sum_{k=0}^{J-1} \lambda_k (1 - a_k^T a_k^*)$ Differentiating L with respect to a_j: $R_{ff} a_j^* = \lambda_i a_j^* \quad \forall_j < J$ necessary condition

Simple recognition

SSD between images, best match wins very expensive, since need to correlate with every image

Principle Component analysis PCA

Steps: standardize data, get Eigenvectors and values from covariance matrix or do SVD, sort Eigenvalues and vectors in descending order get j largest components, construct projection matrix from selected j Eigenvectors transform dataset by multiplying with projection matrix.

TODO

Uses of PCA: lossycompression by keeping only the most important k components. Face recognition eigenfaces and face detection.

Eigenspace matching

Do PCA with mean subtraction and get closest rank-k approximation of database images (eignfaces)

For a new query: normalize, subtract mean (of database) project to subspace then do similarity matching with eigenfaces.

4.1 Fischerfaces:

Find directions where ratio between / within individual variance is maximized. Linearly project to basis where dimension with good signal: noise ratio is maximized.

$W_{\text{opt}} = \arg\max_W \frac{\det(W R_B W^H)}{\det(W R_W W^H)}, R_b = \sum_{i=1}^{c} (\mu_i - \mu)(\mu_i - \mu)^T, R_W = \sum_{i=1}^{c} \sum_{x \in X_i} (x - \mu_i)(x - \mu_i)^T$

5 Pyramids and Wavelets

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

6 Textures

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

7 Augmented Reality

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

8 Drawing Triangles

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

9 Transforms

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

10 Transform Geometry and Textures

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

11 Graphics Pipeline

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

12 Rendering Equation

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

13 Rigging, FK and IK

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

14 Physically-based Animation

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

15 Partial Differential Equations

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

16 Animation

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.