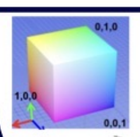
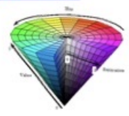
CS 194-26 Midterm Notes — Jeffrey Shen

① Color Spaces





1) RGB : Easy for devices, not perceptual
2) HSV : Hue, Saturation, Value, intuitive
3) L*a*b : "Perceptually uniform" color space

② Point Processing & Filtering : change image range

- Histogram equalization of value channel ↑ constrast

$S = T(r)$ T: cumulative histogram

Aliasing: undersampling causes diff signals shown
↳ counter: sample more, remove high freq

Linear Filtering: transformation, use convolution

Cross-correlation: $H \otimes F = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v] F[i+u, j+v]$

Convolution: $H * F$ cross-correlation where filter flipped

Gaussian Filter: further pixels less weight, low-pass filter

Image Pyramid: Filter → Subsample until min resolution reached, good for search scale

Image Sharpen: $img + \alpha(img - img * g) = img * ((1+\alpha)e - \alpha g)$ ← unit impulse

③ Frequency Domain

Fourier: any univariate func as weighted sum of sin/cos

Fourier Transform: $f(x) \rightarrow$ [Fourier Trans] $\rightarrow F(w)$

$F(w)$ holds $A\sin(wx+\phi)$

- Convolution in spatial dom same as mult in freq domain

④ Pyramid Blending

Laplacian Pyramid: Blending

1) Build Laplacian pyramid LA, LB from images A, B
2) Build Gaussian pyramid GR from region R
3) Combined pyramid:

$$LS(i,j) = GR(I_{ij}) * LA(I_{ij}) + (1 - GR(I_{ij})) * LB(I_{ij})$$

4) Collapse LS pyramid

Matching Filters: methods

1) Zero-mean filter: fastest, not good matching
2) SSD: next fastest, sensitive to overall intensity
3) Normalized cross-coorelation: slowest, invariant to local avg, constrast

Median Filter: robust to outliers

⑤ Image Transformations : change image domain

Types of Parametric Warping

1) Linear Transformations:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Properties:
1) origin → origin 2) lines → lines 3) parallel → parallel

Rotation:
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Scaling:
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Shear:
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & sh_x \\ sh_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- Translation
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$ DoF: 2

- Rigid: translation + rotation DoF: 3
- Similarity: rigid + scaling DoF: 4

2) Affine : linear transformation + translation
1) origin ↛ origin 2) lines → lines 3) parallel → parallel

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$ DoF: 6

3) Projective: affine + projective warps
1) origin ↛ origin 2) lines → lines 3) parallel ↛ parallel
4) ratios ↛ ratios

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$ DoF: 8

⑥ Image Warping & Morphing

Image Morphing: Warping + cross dissolve
- Use points to recover Transformation Matrix
1) Coorespondences at key feature points
2) Define triangular mesh over points
3) Warp each triangle in any shape
   - Inverse Warping: destination to source,
   - Interpolation: color value from neighbors

Triangularization: points to triangles

Delaunay Triangularization: max smallest angles

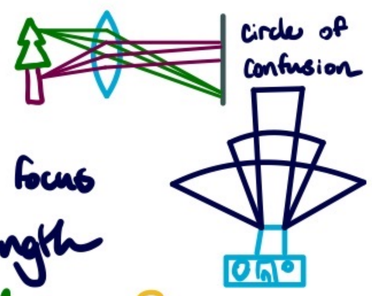
- Can model faces in linear subspace, subsets

Principal Component Analysis: use to find eigenfaces

⑦ Camera

Pinhole Camera: captures pencil of rays through Center of Projection formed on Image Plane

- Smaller aperture makes more clear, too much → diffraction



Focus: shape of lens changes focus

Depth of Field: controlled by aperture
Smaller aperture increases range of focus

Field of View (Zoom): depends on focal length

Exposure: can be adjusted w/ shutter speed

⑧ Homographies & Panoramas

Plenoptic function: $P(\theta, \phi, \lambda, t, V_x, V_y, V_z)$ every view at every moment from every position, at every wavelen

Homography: Projective mapping between pics w/ same center of projection

$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad p' = Hp$$

Least-Squares: use to solve points when noisy $x = (A^T A)^{-1} A^T b$, $Ax = b$

Panoramas: Synthetic wide angle
1) Pick image 2) Warp all other toward it 3) blend

Blending w/ alpha, set overlap $\alpha = 0.5$

Cylindral Projection: map onto cylinder

$(\hat{x},\hat{y},\hat{z}) = \frac{1}{\sqrt{x^2+z^2}}(x,y,z)$

Spherical projection: map onto sphere

$(\hat{x},\hat{y},\hat{z}) = \frac{1}{\sqrt{x^2+y^2+z^2}}(x,y,z)$

$(\sin\theta\cos\phi, \sin\phi, \cos\theta\cos\phi) = (\hat{x},\hat{y},\hat{z})$

Radial Distortion: bending in wide field of view

$\hat{r}^2 = \hat{x}^2+\hat{y}^2$, $\hat{x}' = \hat{x}/(1+k_1\hat{r}^2+k_2\hat{r}^4)$, $\hat{y}' = \hat{y}/(1+k_1\hat{r}^2+k_2\hat{r}^4)$

$x = f\hat{x}'/\hat{z} + x_c$, $y = f\hat{y}'/\hat{z} + y_c$

9 Automatic Image Alignment

1) Direct Alignment: SSD, NCC, bruteforce
2) Feature Based Alignment:
   a) Feature Detection: Interest points
   b) Feature Matching: match across imgs
   c) Compute Image Transformation

Invariant Local Features: local feature coords invariant to translation, rotation, scale

Harris Detector: window has significant change in all directions

- $R = \frac{\det M}{\text{trace } M}$ $\det M = \lambda_1\lambda_2$ $\text{trace } M = \lambda_1+\lambda_2$ find where λ_1,λ_2 are large
- Steps: Gaussian derivatives at pixels, second moment matrix M, corner response R, threshold R, local max

Adaptive Non-maximal Suppression: fixed # points distributed spatially evenly, sort by non-maximal suppression radius

Feature Descriptor: patch around feature &x&

For every patch find most similar patch

- only match SSD(patch1, patch2) < threshold

Lowe's Tricks: look at how much better closest match is compared to second-closest

RANdom SAmple Consensus (RANSAC): find right H

1) Find 4 pts at random
2) Compute homography H
3) Compute inliers where $dist(p_i', Hp_i) < \varepsilon$
4) Keep largest set inliers
5) Recompute least squares H estimate

Optical Flow: estimate motion of pixel, assume color constancy, small motion

Lukas-Kanade Algorithm

1) Estimate velocity at each pixel solving Lukas-Kanade equations $(A^TA)d = A^Tb$
2) Warp H toward I using est flow field
3) Repeat till convergence

10 Visual Texture

Oriented Gaussian Derivatives $Rot_\theta f_1 = G_{\sigma_1}(u)G_{\sigma_2}(v)$ $Rot_\theta f_2 = G'_{\sigma_1}(u)G_{\sigma_2}(v)$

Simoncelli & Portilla: match 2nd order statistics

Image texture as bag of "patches"

11 Feature Learning w/ Neural Nets

Statistical Learning Framework $y = f(x)$

Training Set: labeled examples, estimate f min error

Test Set: apply f to never seen x output predicted value

Validation: same as test, held out

Objective Loss funcs: min loss

Quadratic loss: $l(x_i,y_i;w) = (f(x_i)-y_i)^2$

Log Likelihood: $l(x_i,y_i;w) = -\log P_w(y_i|x_i)$

Perceptron Training Algo

1) initialize w randomly 2) cycle through multiple passes
3) for each x,y $y' = sgn(w\cdot x)$ classify, update $w \leftarrow w+\alpha(y'-y)x$

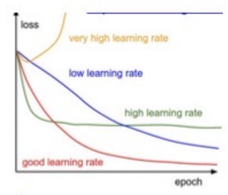
Multi layer perceptrons: add nonlinearity

Sigmoid: $g(t) = \frac{1}{1+e^{-t}}$ Rectified Linear Unit (ReLU): $g(t) = \max(0,t)$

Backpropogation: chain rule $\frac{\partial L}{\partial w^2} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial x^2}\frac{\partial x^2}{\partial w^2}$

12 Convolutional Neural Nets

- Learn multiple filters
- Most memory usage in early convolutional layers, parameters in fully connected layers
- Training diagnosis



Transfer learning: if small dataset retrain classifier/some fully connected

Upsampling: transposed-strided, nearest neighbor, interpolation

13 Conv nets as tool

Depth Prediction ex

Generative Adversial Network (GANs): G synthesizes fake images that fool D, D identify fakes

$\arg\min_G \max_D \mathbb{E}_{x,y}[\log D(x, G(x)) + \log(1-D(x,y))]$

CycleGAN: cycle there and back until same

14 3D Vision: Calibration Stereo

Estimating Depth w/ Stereo

Stereo: shape from "motion" between two views

Need: 1) camera pose "calibration" 2) point correspondences

$v \times u$ pixels $\hat{x}_i$ $\hat{y}_i$

Pixels → 3D location in world

Need: Extrinsics: position of camera w/ respect to world

Intrinsics: how camera maps pt in world to image

Image Plane to Image Sensor Mapping

$[m_x, m_y]$ $u_i = \alpha_x x_i + o_x = \alpha_x f\frac{x_c}{z_c} + o_x$

$u_i = f_x\frac{x_c}{z_c} + o_x$ $u_i = f_y\frac{y_c}{z_c} + o_y$

image coord $\begin{bmatrix}x_i\\y_i\end{bmatrix}$ ← Intrinsics: perspective projection ← camera coord $\begin{bmatrix}x_c\\y_c\\z_c\end{bmatrix}$ ← Extrinsics: coordinate transformation ← world coord $\begin{bmatrix}x_w\\y_w\\z_w\end{bmatrix}$

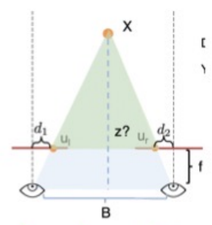
Calibration

take pic of obj known 3D geometry

coorespondences

solve m's least squares

$Z = \frac{fB}{u_l - u_r}$ ← disparity



Dense Correspondence Search

1) for epipolar line,
   for window
   compare on epipolar line, pick min