1 Pinhole camera

- Barrier with a hole and a film
- Focal length - f
- Perspective projection equation $\frac{f}{-y'} = \frac{-z}{y}$
- Apature - size of pinhole
- Lense - focusing element
- Depth of field - distance and size of small blurring
- Field of view - $\varphi = tan^{-1}(\frac{d}{2f})$
- Chromatic aberration - different wavelength refraction
- Spherical aberration - spherical lenses
- Vignetting - edge of camera absorbtion
- Radial distortion - lens imperfections
- Digitalization - discretization, quantization

2 Color spaces

- Additive models *(RGB)* - colors added to black
- Subtractive models *(CYMK)* - colors added to white
- Linear color space - CIE XYZ
  Artificial primaries X, Y, Z
  $x = \frac{X}{X+Y+Z}$, $y = \frac{Y}{X+Y+Z}$, $z = \frac{Z}{X+Y+Z}$, $x + y + z = 1$
  Chromacity is represented using only $[x, y]$
- RGB
- Nonlinear color space - HSV
- Uniform color space - CIE u'v'

3 Basic image processing

Basic process
- Localize
- Describe
- Classify

3.1 Thresholding

Transforms an image into a binary mask
- Single(two) threshold approach
  $F_T[i,j] = \begin{cases} 1, \text{if } T_1 \leq F[i,j] (\leq T_2) \\ 0, \text{otherwise} \end{cases}$
- General approach
  $F_T[i,j] = \begin{cases} 1, \text{if } F[i,j] \in Z \\ 0, \text{otherwise} \end{cases}$
- Global binarization
  Otsu's method
  Minimizes within class variance
  $\sigma^2_{within}(T) = n_1(T)\sigma_1^2(T) + n_2(T)\sigma_2^2(T) \equiv$ maximization of $\sigma^2_{between}(T) = \sigma^2 - \sigma^2_{within}(T) = n_1(T)n_2(T)[\mu_1(T) - \mu_2(T)]$
  Find $T^* = \text{argmax}_T[\sigma^2_{between}(T)]$
- Local binarization
  Estimate local threshold in neighborhood W $T_W = \mu_W + k\sigma_W$ for $k \in [-1, 1]$
- Shade compensation using polynomials

3.2 Morphology

Structuring element $\begin{cases} \text{Fit: all 1's cover 1's in SE} \\ \text{Hit: at least 1 covers a 1 in SE} \end{cases}$
- Erosion $g = f \ominus s$
  $g(x,y) = \begin{cases} 1, \text{if s fits f} \\ 0, otherwise \end{cases}$
- Dilation $g = f \oplus s$
  $g(x,y) = \begin{cases} 1, \text{if s hits f} \\ 0, otherwise \end{cases}$
- Opening $A \circ B = (A \ominus B) \oplus B$ - opens gaps, holes
- Closing $A \bullet B = (A \oplus B) \ominus B$ - closes gaps, holes

3.3 Region descriptors

Labeling components
- 4-8 way connectivity
- Connected components

```
for px in image top to bottom left to right:
 if px == 1:
  if one neighbor top or left:
   label = n_label
  if both neighbors:
   label = n_label if they have same label
   else copy left label and add equivalency
  else:
   label = new label
```

- Describe region
  - Area A
  - Perimeter l
  - Compactness $c = l^2(4\pi A)$
  - Circularity l/c ...

Color similarity between objects
- Average color
- Fit Gaussian distribution
- Histograms - $H(c)$ = number of px with color c
  Robust to translation, scale, partial occlusion
- Intensity normalization
  $r = \frac{R}{R+G+B}$, $g = \frac{G}{R+G+B}$, $b = \frac{B}{R+G+B}$
  Reduce to 2D ($[r, g]$) since $r + g + b = 1$

Distances
- L_2 norm (*Euclidean*) $d(Q,V) = \sqrt{\sum_i (q_i - v_i)^2}$
- $\chi^2(Q,V) = \sum_i \frac{(q_i - v_i)^2}{q_i + v_i}$
- Hellinger $d_{Hell}(Q,V) = \sqrt{1 - \sum_i \sqrt{q_i v_i}}$

3.4 (Non)linear filters

Types of noise
- Salt and pepper
- Impulse noise
- Gaussian noise

Convolution
- Correlation $G = H \otimes F$
  $G[i,j] = \sum_{u=-k}^{k} \sum_{v=-l}^{k} H[u,v]F[i+u, j+v]$
- Convolution $G = H \star F$
  $G[i,j] = \sum_{u=-k}^{k} \sum_{v=-l}^{k} H[u,v]F[i-u, j-v]$
  If $H[-u,-v] = H[u,v]$, then $\otimes \equiv \star$
- Properties
  - Linear - $h \star (\alpha_1 f_1 + \alpha_2 f_2) = \alpha_1(h \star f_1) + \alpha_2(h \star f_2)$
  - Commutative - $f \star g = g \star f$
  - Associative - $(f \star g) \star h = f \star (g \star h)$
    and so also $((f \star b_1) \star b_2) \star b_3 = f \star (b_1 \star b_2 \star b_3)$
  - Derivative - $\frac{\partial}{\partial x}(f \star g) = (\frac{\partial}{\partial x} f) \star g = (\frac{\partial}{\partial x} g) \star f$
- Boundry conditions
  - Crop
  - Bend image around

- replicate edges
- Mirror image

- Image pyramids
  Nyquist theorem - sample the signal by at least $2f$ - Remove high frequencies before sub-sampling
  Gaussian pyramid
  $G_i = (G_{i-1} \star \text{Gaussian}) \downarrow 2$, $G_0 = \text{image}$

4 Edge detection and image gradients

4.1 Image derivatives

Discrete case (images)
$\frac{\partial f(x,y)}{\partial x} = \frac{f(x+1,y)-f(x,y)}{1}$
Gradient magnitude
$\nabla f = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}]$
Direction - $\theta = tan^{-1}(\frac{\partial f}{\partial x} / \frac{\partial f}{\partial y})$
Magnitude - $\|\nabla f\| = \sqrt{(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2}$
Smarter derivative - $\frac{\partial}{\partial x}(I \star G) = I \star (\frac{\partial}{\partial x} G)$

4.2 Canny edge detector

Good edge detector

- Detection - minimizes FP and FN
- Localization - close to true edge
- Specificity - minimize local maxima

Process

1. Calculate $I_\partial = I \star \frac{\partial}{\partial x} G$
2. Calculate θ, $\|\nabla f\|$
3. Non-maxima suppression
4. Trace edges by hysteresis thresholding

4.3 Hough transform

Process

1. For each edge point compute all possible parameters passing through that point
2. For each set of parameters cast a vote
3. Select parameter combinations that receive enough votes

Lines - $x \cos\theta - y \sin\theta = d$
Circles - $(x-a)^2 + (y-b)^2 = r^2$
Extensions

- Use gradient direction for θ
- Use magnitude for voting weight
- Generalized Hough transform

5 Fitting models

Transformation of points $x'_i = f(x_i; \mathbf{p})$

5.1 Least-squares

Minimizing a continuous error function $\tilde{p} = argmin_{\mathbf{p}} E(\mathbf{p})$
$\epsilon_i = f(x_i; \mathbf{p}) - y_i$
$E(\mathbf{p}) = \sum_{i=1}^N \epsilon_i^2$
Strategy

1. Rewrite the cost function $E(\mathbf{p})$ into vector-matrix form
2. $\frac{\partial E(\mathbf{p})}{\partial \mathbf{p}} = 0$; solve for p

Derivative of linear and quadratic form
$\frac{\partial \mathbf{A^T p}}{\partial \mathbf{p}} = \mathbf{A^T}$, $\frac{\partial \mathbf{p^T A p}}{\partial \mathbf{p}} = \mathbf{2Ap}$

5.2 Normal equations

Rewrite the error into $\mathbf{Ap} = \mathbf{b}$
Solve by $\mathbf{p} = \mathbf{A}^\dagger \mathbf{b} = (\mathbf{A^T A})^{-1} \mathbf{A^T b}$
Weighted least squares
$E(\mathbf{p}) = \sum_{i=1}^N w_i \epsilon_i^2$
Solve by $\mathbf{p} = (\mathbf{A^T W A})^{-1} \mathbf{A^T W b}$

5.3 Homogenus systems

Constrained least squares $Ap = \lambda p \rightarrow \mathbf{Ap} = \mathbf{0}$ with $\|p\|^2 = 1$
Solve by $svd(A)$; $\mathbf{p}$ is eig. vector with smallest eig. value

5.4 Nonlinear cost function

- Gradient descend
- Newtons method
- Levenberg-Marquardt ...

5.5 RANSAC

Ransac loop

1. Randomly select s correspondences
2. Fit model parameters
3. Count projected inliers
4. Remember the optimal parameters

e - probability of outlier
s - minimal number of correspondences to fit a model
p - probability of drawing all inliers at least once
Number of iterations $N = \frac{\log(1-p)}{\log(1-(1-e)^s)}$

6 Keypoints and correspondences

6.1 Keypoint detection

Harris corner detector
$M = \begin{bmatrix} G(\sigma) \star I_x^2 & G(\sigma) \star I_x I_y \\ G(\sigma) \star I_x I_y & G(\sigma) \star I_y^2 \end{bmatrix} = R \begin{bmatrix} \lambda_{max} & 0 \\ 0 & \lambda_{min} \end{bmatrix} R^T$
$det(M) - \alpha trace^2(M) > t$, $0.04 < \alpha < 0.06$
$det(M) = AB - C^2$, $trace(M) = A + B$
$CRF(I) = det(M) - \alpha trace(M)$
Process

1. Image derivatives
2. Squared derivatives
3. Gaussian filtered squared derivatives
4. Corner response function
5. Non-maxima suppression

Hessian corner detector
$Hessian(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$
$CRF(I) = det(Hessian(I)) = I_{xx} I_{yy} - I_{xy}^2$
Process

1. Image derivatives
2. Second order derivatives
3. Corner response function
4. Non-maxima suppression

Laplacian of Gaussian
Used as scale response function
$LoG = \nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$
Process

1. Laplacian pyramid
2. Scale space non-maxima suppression

Difference of Gaussian
$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$
Process

1. Gaussian pyramid (faster because of down-sampling)

2. DoG pyramid from Gaussian
3. Scale space non-maxima suppression
4. Remove low contrast points
5. Remove points detected at edges

6.2 Local descriptors

- Vector of region intensities
- SIFT

6.3 SIFT

1. Split region in 4x4 cells
2. Calculate gradient
3. 8 bin histogram of gradient weighted by magnitude and region center
4. Stack histograms and normalize

Rotation invariance
36 bins by angle, rotate gradients using dominant rotation, create descriptor for every orientation with magnitude at least 80% of maximum.
Affine adaptation
Start with circular window, estimate new window using the covariance matrix of current window. Iterate until convergence.
Rotate $R = U^{-1}$ and scale $S^{-1/2}$ from window ellipse $\Sigma = USU^T$, calculate descriptor using affine adapted region.

6.4 Correspondences

- Find most similar descriptor
- Keep only symmetric
- Keep only distinctive (ratio to second most similar)

7 Cameras and stereo systems

Projection: extrinsic *world* $\rightarrow$ *camera*, intrinsic *camera* $\rightarrow$ *image*

7.1 Pinhole camera model

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} fX/Z \\ fY/Z \\ 1 \end{bmatrix} \quad K = \begin{bmatrix} a_x & s & x_0 \\ 0 & a_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \quad P_0 = K[I|0]$$

Principal point $[p_x, p_y]$, focal length f, pixels per meter $[m_x, m_y]$, skew s
$a_x = fm_x$, $a_y = fm_y$, $x_0 = p_x m_x$, $y_0 = p_y m_y$
$\tilde{X}_{cam} = R(\tilde{X} - \tilde{C}) = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} X$, Camera origin in w.c.s $\tilde{C}$
$P = K[R|t], t = -R\tilde{C}$
Nonlinearity correction using polynomial
$\tilde{x} = x_d + (x_d - c_x)(K_1\rho^2 + K_2\rho^4 + \dots)$, $\tilde{y} = \dots$
Degrees of freedom
$[p_x, p_y]$: 2, f: 1, $[m_x \equiv_{rectangular} m_y]$: 1(2), s: 1, R: 3, t: 3

7.2 Homography

$w\mathbf{x}' = \mathbf{Hx}$
Direct linear transformation
$[a_\times] \equiv \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$; $\mathbf{x_i'} \times \mathbf{Hx_i} = 0$; $\mathbf{Ah} = \mathbf{0}$
Preconditioning
$T_{pre} = \begin{bmatrix} a & 0 & c \\ 0 & b & d \\ 0 & 0 & 1 \end{bmatrix}$; $\tilde{x} = T_{pre}x$; $\tilde{x}_i' \times \tilde{H}\tilde{x}_i = 0$; $H = T_{pre}^{'-1}\tilde{H}T_{pre}$
Set a, b, c, d so that mean $\tilde{x}_i$ is 0 and variance is 1

7.3 Vanishing points

$$v = \begin{bmatrix} fX_D/Z_D \\ fY_D/Z_D \end{bmatrix}$$

7.4 Calibration

Estimate $\mathbf{P}$ from a known calibration object.

- Using DLT and preconditioning $\mathbf{x_i'} \times \mathbf{Px_i} = 0$ and decompose to $K[R|t]$
- Using minimization of error
  $\varepsilon_i = \begin{bmatrix} \varepsilon_{xi} \\ \varepsilon_{yi} \end{bmatrix} = (x_i - PX_i)$ $E(P) = \sum_{i=1}^{N} \varepsilon_i^T \varepsilon_i$
- Multiplane calibration

7.5 Triangulation

Using DLT with $[x_{1\times}]P_1X = 0$ and $[x_{2\times}]P_2X = 0$, then minimize sum of re-projection errors using iterative algorithm $E(X) = d^2(x_1, P_1X) + d^2(x_2, P_2X)$

7.6 Epipolar geometry

Epipolar constraint $X^T([T_\times]RX') = 0$; $x^TEx' = 0$
Essential matrix $E = [T_\times]R$ constrains x and x' in meters
Epipolar line vector
$l' = E^Tx$; $l = Ex'$
Fundamental matrix constrains $\hat{x}$ and $\hat{x}'$ in pixels
$F = K^{-T}EK^{'-1}$
Epipolar lines $\hat{l}' = F^Tx$; $\hat{l} = Fx'$
Epipole $Fe' = 0$, $F^Te = 0$

7.7 Simple stereo

Baseline b_x, focal length f
3D from disparity
$X = x_L\frac{b_x}{d}$, $Y = y_L\frac{b_x}{d}$, $Z = f\frac{b_x}{d}$
Global disparity optimization
Globally consistent solution
$E_{data}(d_i) = e^{-similarity(d_i)}$
$E(d_i) = E_{data}(d_i) + \lambda E_S(d_i)$
Semi global block matching - apply line based optimization across several directions

7.8 Structure from motion

1. Find keypoint correspondences
2. estimate F (weak calibration)
3. get E from F and K_1, K_2
4. get $[R|t]$
5. triangluation

Normalized 8-point algorithm

1. Precondition with $\mu = 0$ and $\sigma = \sqrt{2}$
2. Create homogeneous system using correspondences and epipolar constraint
3. Minimize $\sum_{i=1}^{N}(x_i^TFX_i')$ and solve
4. Set last λ to 0 and reconstruct
5. Transform to original units $F = T^{'T}\tilde{F}T$

Fundamental matrix estimation

1. Find keypoint and correspondences using proximity constraint
2. filter correspondences by visual similarity
3. apply RANSAC with 8-point algorithm and epipolar constraint pruning

7.9 Active stereo

Project light patterns over the object
Multi band triangulation: Assume smooth surface, project color bands

8 Feature learning

8.1 Natural linear coordinate systems

Principal component analysis
$\Sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)(x_u - \mu)^T = \frac{1}{N}X_d X_d^T, X_d = X - \mu$
Maximize $u^T \Sigma u \rightarrow \Sigma u = \lambda u$, solutions are $U =$ eig. vectors of Σ.
Project data to PCA c.s. $y_i = U^T(x_i - \mu)$
Project data from PCA c.s. $x_i = Uy_i + \mu$
Dual PCA
If sample dimension $M >$ number of samples N
$\Sigma' = \frac{1}{N}X_d^T X_d$
$u_i = \frac{X_d u_i'}{\sqrt{N\lambda_i'}}$
Classification by subspace recognition
If window contains a trained subspace the reconstruction will work well. $\|\tilde{x}_i - x_i\|^2 < \theta$
Linear discriminant analysis
$S_W = \sum_{i=1}^{c}\sum_j (x_j^{(i)} - \mu_i)(x_j^{(i)} - \mu_i)^T$
$S_B = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T$
Maximize $J(w) = \frac{w^T S_b w}{w^T S_w w} \rightarrow S_w^{-1}S_b w = \lambda w$, solutions are W = first $c-1$ eig. vectors of $S_w^{-1}S_b w$.

8.2 Nonlinear hand-crafted transforms

Histogram of gradients
1. Calculate gradient
2. Calculate HOG in 8x8 blocks and normalize, weighted by magnitude
3. Train a classifier using support vector machine

8.3 Feature selection

Viola-Jones face detection
Boosting (Adaboost)
- Strong classifier from many weak classifiers
  $h(x) = sign(\sum_{t=1}^{T}\alpha_t h_t(x))$, classifier weight α_t, weak classifier $h_t(x)$
- Weak classifiers using sum of region intensities with integral images $\Sigma(R) = \int_{\ulcorner} + \int_{\lrcorner} - \int_{\llcorner} - \int_{\urcorner}$
- Using cascade of classifiers to reject obvious windows

Region proposals for selective search
Can be used by slow classifiers, hierarchical segmentation

8.4 End-to-end feature & classifier learning

Convolutional neural networks
Feature extraction
- Convolutional layers
- Nonlinearity (RELU)
- Pooling layers

Classifier
- Multi-layer perceptron

Region based CNN evolution
- Slow R-CNN - processes every region proposal through the whole CNN to classify
- Fast R-CNN - process whole image to extract features and join with region proposals to classify
- Faster R-CNN - generate region proposals using extracted features network, multi-scale feature extraction
- Mask R-CNN - Use box regression and additional MLP to create the segmentation mask

9 Keypoint based recognition

9.1 Bag of words models

1. Feature detection and representation
   Use SIFT and affine adaptation, collect all descriptors
2. Dictionary construction
   Cluster descriptors into clusters (K-means) - cluster center is a word
3. Image representation
   Detect words in training images and build word histograms (BOWs)
4. Build a classifier
   Use histograms to make a classifier
5. Recognition
   Extract BOWs and apply them to the classifier

9.2 Detection by RANSAC and GHT

Detection by RANSAC
1. Represent model using affine deformation invariant parts
2. Detect parts in image
3. Use RANSAC with parts and try to fit a good model

Generalized Hough transform
1. Index descriptors
2. Apply GHT to obtain detections, each feature casts a vote into the Hough space
3. Refine detection using affine transform