Technion Institute of technology Israel Amit Levi

Morphology

Erosion(count), Dilation(edge), Opening(insulate), Closing(closed), Thin (skeleton), Thickening (convex)

$$\mathrm{Opening(I)} = \mathrm{Dilation(Erosion(I))}$$
$$\mathrm{Closing(I)} = \mathrm{Erosion(Dilation(I))}$$
$$\mathrm{Thinning(I, SEs)} = \mathrm{I} - \mathrm{Hit\&Miss(I, SEs)}$$
$$\mathrm{Thickening(I, SEs)} = \mathrm{I} + \mathrm{Hit\&Miss(I, SEs)}$$

Texture

Co-occurrence matrix: $C_d(i,j)$ is number of p s.t. $I(p) = i$ and $I(p+d) = j$.

Alg. Histogram of Textons:

1. L: Characterize neigherhood by vector.
2. L: Run K-means clustering. Every cluster is a textone.
3. T: Characterize neigherhood by vector.
4. T: Assign to the nearest cluster.
5. T: Calculate hist. of textons.

Efros & Leung for synthesis:

1. Search the input image for all similar neighborhoods.
2. pick one match at random.

Segmentation

Alg. 1 – Thresholding: maybe local thresholding.

Alg. 2 – K-means: to find local min m_k:

1. Randomly initialize $m_k = \{x_{i_k}\}$.
2. Assign $k(x_i) = \arg\min_k \|x_i - m_k\|$.
3. Assign $m_k = \frac{\sum_{x_i \in S_k} x_i}{|S_k|}$ (cluster center).
4. Repeat 2+3 until convergence.
5. $m_k = \arg\min_x \sum_{x_i \in S_k} \|x_i - x\|^2$

Alg. 3 – EM:

$$P(x) = \sum_k \alpha_k P_k(x) = \sum_k \alpha_k \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\right]$$

E: $\gamma_k(X_i) = \frac{\alpha_k P_k(X_i)}{\sum_j \alpha_j P_j(X_i)}$, M: $\mu_k = \frac{\sum_i \gamma_k(X_i) X_i}{\sum_i \gamma_k(X_i)}$

Alg. 4 –

1. Specify a weighted graph.
2. Initialize clusters as single elements
3. Merge closest clusters until #clus. = K

if clus. dist = d(U,V) this Kruskal for MST.

Perspective projection

$$\begin{pmatrix} x_{im} \\ y_{im} \\ 1 \end{pmatrix} \sim \begin{pmatrix} -f/s_x & 0 & o_x \\ 0 & -f/s_y & o_y \\ 0 & 0 & 1 \end{pmatrix} (I_3 | 0_{3\times1})_{3\times4} \cdot \begin{pmatrix} R^T & -R^T O \\ 0 & 1 \end{pmatrix}_{4\times4} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Hough Transformation

1. For each point p, define $H(\theta)$ as distance from the line which pass in p in angle θ.
2. Draw the respective curves.
3. Seek for intersection of curves.

Edge Detection

Matched filter: $N \sim \mathcal{N}(0,\sigma^2), X = [S_e \mid S_b] + N$. We max $C^T(S_e - S_b)$ with $C = \frac{S_e - S_b}{\|S_e - S_b\|}$ and we look linear map $R = C^T X$.

Gradient approach: big gradient = edge. We take 3x3 neighborhood $[I_1, I_2, I_3 ; I_4, I_5, I_6 ; I_7, I_8, I_9]$. and $H\theta \approx I(:)$ when $\theta^* = (W^T W)^{-1} W I(:)$.

Smoothing: By convolution with gaussian G and then we use gradient (for find edges). 1st der. is max when 2nd der. is zero, i.e. Laplacian is zero.

Discrete Laplacian: [0 1 0 ; 1 -4 1 ; 0 1 0]

Canny Edge Detector:

1. Apply Gaussian smoothing
2. Calculate Gradient magnitude\direction
3. Init: a point s.t. GradMag > Th_high
4. Thin: check neighbors in Grad direction and choose maximum
5. Next: pixel neighbor in tangent direct.
6. If (GradMag > Th_low) continue to (4), Else continue to (3)

Pyramids

Smoothing: $L(p_i, t+1) = \frac{1}{|N_i|} \sum_{p_j \in N_i} L(p_j, t)$

Gaussian Pyramid: $g_\sigma = \frac{1}{2\pi\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$

$$DoG = g_{k\sigma}(p) - g_\sigma(p) \approx \sigma^2 \nabla g_\sigma$$
$$I^{(n+1)} = (g * I^{(n)}) \downarrow, LoG^{(n)} \approx I^{(n)} - g * I^{(n)}$$

Pyramid Blending: Calculation in point p:

$$L_S^{(k)} = G_M^{(k)} \cdot L_A^{(k)} + (1 - G_M^{(k)}) \cdot L_B^{(k)}$$

Learning

Ideal Risk: $R(\phi) = \mathbb{E}\{L(\phi(x), y)\}$

Emp. Risk:

$$R_{emp}(\phi) = \frac{1}{m} \sum_{i=1}^m L_i(\phi(x_i), y_i)$$

Softmax Loss:

$$\Pr[Y = k_0 | X = x] = \frac{e^{s_{k_0}}}{\sum_{k=1}^m e^{s_k}}$$

Cross-Entropy Loss:

$$L_i = -\log \Pr[Y = y_i | X = x_i]$$

General Loss + Regularization:

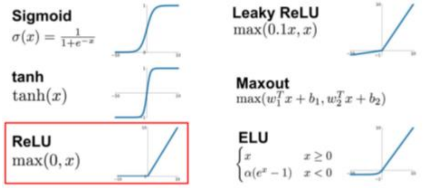
$$L(W) = \frac{1}{m} \sum_{i=1}^m L_i(\phi(x_i), y_i) + \lambda R(W)$$

Gradient Descent: to find local min of f:

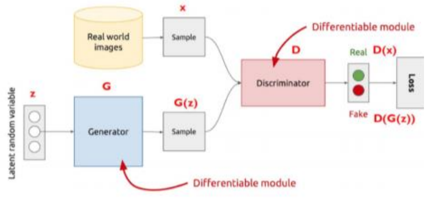
$$x^{(k+1)} = x^{(k)} - t_{k+1} \nabla f(x^{(k)})$$

Gradient of Loss function (or SGD) is to use GD to find local min of $L(W)$.

Activations:



GANs

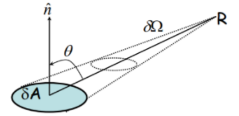


$$\min_G \max_D V(D,G)$$
$$V(D,G) = \mathbb{E}_{x\sim p(x)}[\log D(x)] + [\mathbb{E}_{z\sim q(z)}[\log(1 - D(G(z))]$$

Photometry

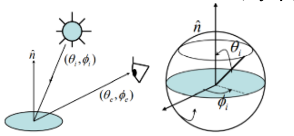
Irradiance: I per area $[Wm^{-2}]$

Radiance: I per area, s. angle $[Wm^{-2}sr^{-1}]$

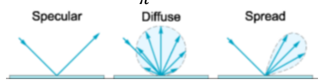


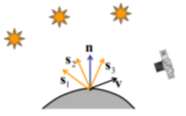
Solid Angle: $\delta\Omega = \frac{\delta A \cos\theta}{R^2}$

$$BRDF(\theta_i, \varphi_i, \theta_e, \varphi_e) = \frac{\delta L(\theta_e, \varphi_e)}{\delta E(\theta_i, \varphi_i)}$$



Lambert: $BRDF = \frac{\rho_0}{\pi}$, $E(\theta_i, \varphi_i) = E_0 \cos\theta_i$



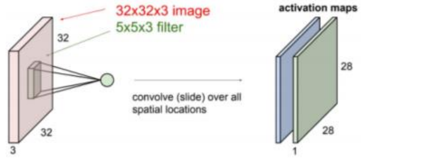


$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \rho \begin{bmatrix} \mathbf{s}_1^T \\ \mathbf{s}_2^T \\ \mathbf{s}_3^T \end{bmatrix} \mathbf{n}$$

$\frac{\partial}{\partial \underline{x}}(\underline{a}^T \underline{x}) = \frac{\partial}{\partial \underline{x}}(\underline{x}^T \underline{a}) = \underline{a}$	$\frac{\partial z}{\partial x} = \frac{\partial y}{\partial x}\frac{\partial z}{\partial y}$
$\frac{\partial}{\partial \underline{x}}(\underline{x}^T \underline{A} \underline{x}) = (\underline{A}^T + \underline{A})\underline{x}$	$\frac{\partial}{\partial \underline{x}} \underline{A}^T \underline{x} = \frac{\partial}{\partial \underline{x}} \underline{x}^T \underline{A} = \underline{A}$

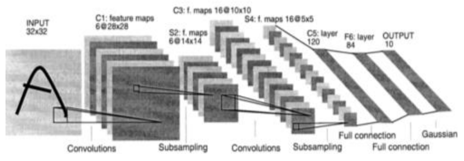
CNN

Convolution (Image, filter sizes, num filters, padding, stride)



Pooling (in [m,n,k]) -> out [m/2,n/2,k]

LeNet5 Architecture:



Technion Institute of technology Israel Amit Levi

Alg. 1 – Intensity subspaces

Alg. NN: for every subimage find:

$$SSD(\text{model}, \text{image})$$

and choose which has the smallest.

Alg. PCA: we find ($\bar{x}$ is mean):

$$var(v) = \sum_x \|(x-\bar{x})^T v\|^2 = v^T cov(x) v$$

and $cov(x) = \sum_x (x-\bar{x})(x-\bar{x})^T$.

max $var(v)$ means large e. value.

min $var(v)$ means small e. value.

$W - m$ high e. vec, $y = W^T x + \bar{x}$:

Alg. low dim: u is e. vector of TT^T iff Tu is e. vector of $T^T T$.

Alg. with SVD: $cov(x) = \frac{1}{n} XX^T$ so:

$$[e.value]_{XX^T,i} = \frac{1}{n}[s.value]_{X,i}$$

Alg. Fisherfaces: $\max \frac{|W^T S_B W|}{|W^T S_W W|}$ where

$$S_W = \sum_{k=1}^K \sum_{x \in c_k} (x-\mu_k)(x-\mu_k)^T$$

$$S_B = \sum_{k=1}^K |N_k| (\mu_k - \mu)(\mu_k - \mu)^T$$

We need find $S_B W = S_W W D$. We can find e. vectors of $(S_W)^{-1} S_B$.

Alg. 2 – Invariants (for binary)

Alg.: Use some features as area, #holes and moment [m] in general:

$$m_{pq} = \sum_{(x,y)} x^p y^q I_{xy}$$

which invariant to isometrics.

invr. Translation – center of mass $\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}$ and cent. [m]:

$$\mu_{pq} = \sum_{(x,y)} (x-\bar{x})^p (y-\bar{y})^q I_{xy}$$

invr. Rotation – e. values of matrix $[\mu_{20}\ \mu_{11}\ ;\ \mu_{11}\ \mu_{02}]$ as in Harris.

Alg. 7 – deformable part

combinate tree based spatial model, HoG in 2-scales and whole object + Parts. Locate in start model, score is appearance – deformation. Training by SVM.

Alg. 3 – Geometric alignment

Alg. RANSAC: if there are $W\%$ inliers:

1. Randomly select subset in size n.
2. Calculate the model parameters p.
3. Assign a score (e.g. #inliers) to p.
4. Re-iterate k times.
5. Choose p with best score.

success of q need $k > \frac{\log(1-q)}{\log(1-W^n)}$

Alg. 4 – Hist. of Gradients (HoG)

In each 8x8 block calculate a histogram of gradient orientations, overlapping blocks, different

Alg. 6 – Viola Jones

Strong classifier as (~200) weak classifiers with adaboost:

$$h(x) = \begin{cases} 1 & \frac{\sum a_k h_k(x)}{\sum a_k} > \frac{1}{2} \\ 0 & else \end{cases}$$

Optical Flow

Assumptions: const brightness, motion is small, const flow field.

Alg. Lucas-Kanade: Solve by LS:

$$\begin{bmatrix} I_{x_1} & I_{y_1} \\ I_{x_2} & I_{y_2} \\ \vdots & \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_{t_1} \\ I_{t_2} \\ \vdots \end{bmatrix}$$

Improvements. Asmp1 – use other property, in homogenic regions (it's not work) the flow is smooth, combine segmentation, Asmp2 – multiscale.

Normal (sensitive illumination):

$$U_{ac} = \left(\frac{1}{N(p)} \sum_{p' \in N(p)} U(p')\right) \cdot \bar{1}$$

$$\hat{U}(p) = \frac{U(p) - U_{ac}}{\|U(p) - U_{ac}\|}$$

$$NCC(U,V) = \sum_p \hat{U}(p) \hat{V}(p)$$

Gradient. Orientation [O]:

$\theta_U(p) = \angle \nabla U(p)$:

$$O - SSD = \sum_p (\theta_U(p) - \theta_V(p) \bmod \pi)^2$$

$$\chi^2(h_i, h_j) = \frac{1}{2} \sum_{m=1}^{K} \frac{[h_i(m) - h_j(m)]^2}{h_i(m) + h_j(m)}$$

Alg. 5 - Bag of features

Alg. Basic:

1. Extract features – extract to windows and represent each one (HoG or PCA).
2. Learn visual words (=VWs) – K-means.
3. Quantize features by VWs.
4. Represent image as a hist. of VWs.
5. Train a classifier e.g. SVM.

Alg. with Spatial Considerations: Training:

1. Quantize to words
2. For every example:
   a. divide ROI into spatial pyramid.
   b. find histogram for every cell.
3. Train a learning algorithm.

In test we build representation and classify.

Feature Points

$$SelfDS(x_0, u) = \sum_{x_i \in N(x_0)} w_i \big(I(x_i) - I(x_i + u)\big)^2 = u^T A u = \lambda_1 \cos^2\theta + \lambda_2 \sin^2\theta$$

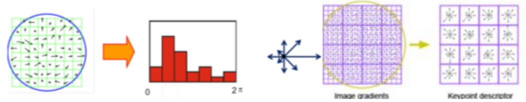
where $A = \begin{pmatrix} \sum w_i I_x^2(x_i) & \sum w_i I_x(x_i) I_y(x_i) \\ \sum w_i I_x(x_i) I_y(x_i) & \sum w_i I_y^2(x_i) \end{pmatrix}$.

Harris Alg.: for each point find (*) and then threshold and local max. (*): V1 λ_2, V2 $\lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$.

SSD(U, V): $\sum_{p \in neigh} \big(U(p) - V(p)\big)^2$

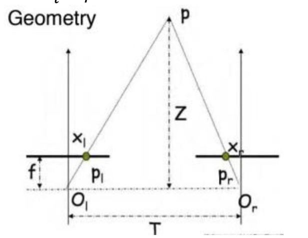
Alg. SIFT:

1. Find local maxima of DoG
2. For every detected point, correct for pose. Find dominant orientation by:
   a. Find gradients at a neighborhood
   b. Calculate weighted orientation histogram
   c. Choose the highest value as orientation
3. Calculate the descriptor:
   a. divide patch into 4x4 = 16 cells
   b. For each cell, compute weighted 8-bin histogram of g. orientation.
   c. Put into a 4x4x8 = 128 dim descriptor.
   d. Make high values uniform and normalize.



Epipolar Geometry

Prop.: $Z = f\frac{T}{x_l - x_r}$ where $x_l - x_r$ is disparity.



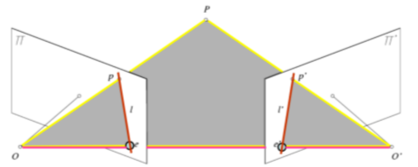
Camera Calibration. We have:

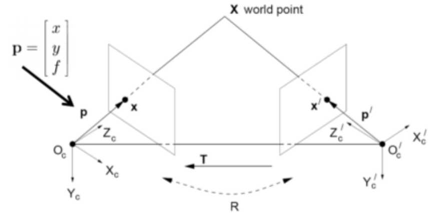
$$0 = (M_1 - x_{im} M_3) P_w = (M_2 - y_{im} M_3) P_w$$

$$\begin{pmatrix} X_w & Y_w & Z_w & 1 & 0 & 0 & 0 & 0 & -x_{im}X_w & -x_{im}Y_w & -x_{im}Z_w & -x_{im} \\ 0 & 0 & 0 & 0 & X_w & Y_w & Z_w & 1 & -y_{im}X_w & -y_{im}Y_w & -y_{im}Z_w & -y_{im} \end{pmatrix} \begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Every point gives two constraints on M

finding by SVD (e. vector for min e. value).





$X' = RX + T$ so $X' \cdot (T \times RX) = 0$.

Essential Matrix: $E = T_\times R$, rank=2, free=5.

Epipolar constrain: $p'^T E p = 0$ or $X'^T E X = 0$.

Fundamental Matrix: $F = K^{-T} E K^{-1}$, rank=2, free=7 and we have $\bar{p}'^T F \bar{p} = 0$.

Alg. for finding F: collect 8+ constrains, normalize, solve using SVD, rearrange and find a rank 2 approximation. A constrain is:

$$[u' \ \ v' \ \ 1] F [u \ \ v \ \ 1]^T = 0$$