Assignment 3 CS-512 Computer Vision

Fall 2018

Diego Martin Crespo

A20432558

Question 1: Corner Detection

(a) A corner is a local place in the image where the gradient vectors take more than one orientation. The steps for detecting a corner in a local window are:

1) Find the correlation matrix in the window
2) Compute eigenvalues of the matrix
3) Check if $\lambda_1, \lambda_2 > \tau$

(b) PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observation of possibly correlated variables into a set of linearly uncorrelated variables. The number of principal components is equal or less than the smaller number of original values.

(c) $$C = \sum_{i=1}^{n} P_i P_i^t = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \begin{bmatrix} x_i & y_i \end{bmatrix} = \begin{bmatrix} \sum x_i^2 & \sum x_i y_i \\ \sum x_i y_i & \sum y_i^2 \end{bmatrix} = \begin{bmatrix} 4 & 6 \\ 6 & 44 \end{bmatrix}$$

(d) $\lambda_1, \lambda_2 > \tau$ or $\lambda_1, \lambda_2 - k(\lambda_1 + \lambda_2)^2 > \tau$

(e) 1) Compute the eigenvalues λ_1, λ_2 for correlation matrix
2) Sort the pixels based on eigenvalues
3) Start from the top selecting the highest value
4) Delete corners in vicinity of selected corner
5) Stop when you have selected a % of the total

(f) Because it uses $C(C) = \det(C) - k\,tr^2(C)$ $\begin{cases} \det(C) = \lambda_1 * \lambda_2 \\ k\,tr^2(C) = k(\lambda_1 + \lambda_2) \end{cases}$

(g) Project the points onto the edges hypothesis and choose the one with minimal projection. $\sum_i U_I(P_i) U_I(P_i)^T P_i = \sum_i U_I(P_i) U(P_i)^T P$

(h) local object appearance and shape in an image can be described by the distribution, intensity of gradient or edge directions. To perform HOG:

1) Divide image in cells
2) Compute a histogram of gradient direction for each pixel in each cell
3) Compute HOG of each window and concatenate the histograms.

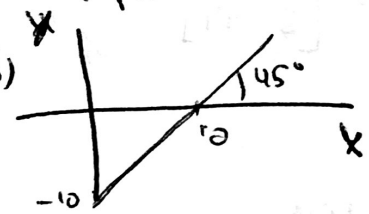
As it operates in local windows it is invariant to geometric and photometric transformations, but not to object orientation.

(i) 1) Create internal representations of the original image to ensure scale invariance
2) Use LoG to find keypoints
3) Eliminate bad keypoints
4) Compute orientation for each keypoint
5) Scale and rotation invariance

Question 2: Line detection

(a) If you use the slope to y-intercept to calculate "a" and "b" in $y = ax + b$ you obtain values $a \in (-\infty, \infty)$ and $b \in (-\infty, \infty)$, this is a problem. Therefore the implicit line equation should be used instead.

(b)



$ax + by + c = 0$

$x = 10,\ y = 0 \rightarrow 10a + c = 0$

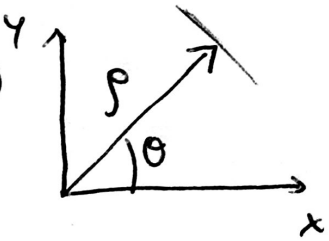
$x = 0,\ y = -10 \rightarrow -10b + c = 0$

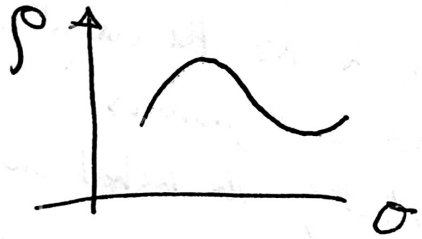
$b = -a$

$a = 10,\ b = -10,\ c = -100$

$10x - 10y - 100 = 0$

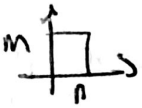
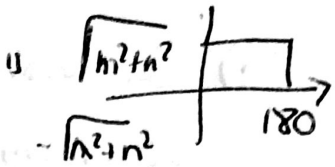
(c)





(d) Each point defines a line in the parameter space. Points on the same line in the x y space define lines. In the parameter space which all intercept in one point. This point define the parameters for line in x y space we are looking for. Steps:

1) Detect Edges
2) Map edge points to hough space and store it
3) Yield stored points in lines of infinite lenght
4) Convert infinite lines to finite line
5) Find intersection

(e) Instead of box $m \times n$  the box size in the parameter is 

(f) Instead of scaling from 0 to 180, it is scaled from $\theta - \Delta$ to $\theta - \Delta$

(g) The parameter is tridimensional (a, b, r)

Question 3: Model Fitting

(a) It does not fit all data with the same precision. Lines with big slopes can not be fitted.

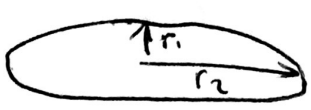
(b) $\begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} [x \ y \ 1] \to x + 2y + 2 = 0$

(c) $E(l) = \sum_{i=1}^{n} (l^t x_i)^2$, Solve $\nabla E(l) = 0$. Compute matrix and find parts to the equations.

(d) $\begin{bmatrix} \Sigma x_i^2 & \Sigma x_i y_i & \Sigma x_i \\ \Sigma x_i y_i & \Sigma y_i^2 & \Sigma y_i \\ \Sigma x_i & \Sigma y_i & n \end{bmatrix} = \begin{bmatrix} 5 & 15 & 3 \\ 15 & 46 & 10 \\ 3 & 10 & 3 \end{bmatrix}$

(e) $ax^2 + bxy + cy^2 + dx + ey + f = 0$, Elipse $\Rightarrow$ $b^2 - 4ac < 0$

(f) $\left.\begin{array}{l} f(x^*) = 0 \\ (p - x^*)^T \begin{pmatrix} \partial f / \partial y \\ \partial f / \partial x \end{pmatrix} = 0 \end{array}\right\}$ Need to solve this



$\frac{d_1}{d_1 + r_1} > \frac{d_2}{d_2 + r_2}$ Points closer to the short axis affect more the fitting

(g) $E(l) = \sum_{i=1}^{n} \frac{|f(p_i, l)|}{|\nabla f(p_i, l)|}$ The problem is that there is no explicit solution to equal the gradient to 0. To calculate it it is used the gradient descent.

(h) $E(\phi(s)) = \int_{\phi(s)} (\alpha(s) E_{continuity} + \beta(s) E_{curvature} + \gamma(s) E_{image}) ds$

- $\alpha(s)$, $\beta(s)$, $\gamma(s)$ are coefficients
- $E_{continuity}$, $E_{curvature}$, E_{image} are energy terms
- $\alpha(s) E_{continuity} + \beta(s) E_{curvature}$ are internal parameters
- $\gamma(s) I_{mage}$ is external parameter

i) In Continuous Space:

- Elasticity $= \left|\frac{\partial \phi}{\partial s}\right|^2$
- Curvature $= \left|\frac{\partial^2 \phi}{\partial s^2}\right|^2$
- Energy $= -|\nabla I|^2$

In Discrete Space:

- Elasticity $= \sum_{i=1}^{n} |P_i - P_{i-1}|^2$
- Curvature $= \sum_{i=1}^{n} |(P_{i+1} - P_i) - (P_i - P_{i-1})|$
- Energy $= \sum |P_{i+1} - 2P_i - P_{i-1}|^2$

ii) Using a threshold if $(P_{i+1} - 2P_i + P_{i-1}) > \tau \Rightarrow \beta = 0$