Assignment 3 Answers

1 a) Corners are the points of interest. The basic principle of detecting the corners is that: the gradient of direction vector are high in ~~both the~~ more than one direction for the corner.

So, the algorithm detects a corner in local window by below steps:

1. Finding correlation matrix in local neighborhood.
2. Find the eigen value of the correlation matrix
3. Check if $ev_1 \cdot ev_2 > \tau$, where ev_1 & ev_2 are the highest eigen value. if $ev_1 \cdot ev_2 > \tau$, detect corner.

In the local neighborhood, if there is more than 1 orientation (i.e, $ev_1 \cdot ev_2 > \tau$, two eigen value are big enough) then we detect a corner. else if there is only 1 orientation we detect an edge.

b) Principal Component Analysis finds the principal detections of gradient orientation in a local path by finding the direction to minimize projection of all points in the local patch.

$$E(v) = \alpha \sum_{i=1}^{n} (P_i \cdot v)^2$$

where $\{P_i\}_{i=1}^{n}$ are n points given in the local path & $E(v)$ is the sum of projections of all P_i's onto direction of V

$$v^* = \underset{v}{\arg\min}\, E(v)$$

$$\Delta E(v) = 2AV \; ; \; A = \sum_{i=n}^{n} P_i^T P_i \text{ is the correlation matrix.}$$

Solve. $AV = 0$; solution is the eigen vector to zero eigen value. of the correlation matrix A ; as it has the lowest projection of all points onto V

c) Correlation matrix $A = \sum_{i=1}^{n} P_i^T P = \begin{bmatrix} \Sigma x_i^2 & \Sigma x_i y_i \\ \Sigma y_i x_i & \Sigma y_i^2 \end{bmatrix}$

$\begin{bmatrix}0\\0\end{bmatrix}[0\ 0] = \begin{bmatrix}0&0\\0&0\end{bmatrix}$ $\quad \begin{bmatrix}0\\4\end{bmatrix}[0\ 4] = \begin{bmatrix}0&0\\0&16\end{bmatrix}$ $\quad \begin{bmatrix}1\\3\end{bmatrix}[1\ 3] = \begin{bmatrix}1&3\\3&9\end{bmatrix}$

$\begin{bmatrix}0\\1\end{bmatrix}[0\ 1] = \begin{bmatrix}0&0\\0&1\end{bmatrix}$ $\quad \begin{bmatrix}1\\0\end{bmatrix}[1\ 0] = \begin{bmatrix}1&0\\0&0\end{bmatrix}$ $\quad A = \begin{bmatrix}4&6\\6&4\end{bmatrix}$

$\begin{bmatrix}0\\2\end{bmatrix}[0\ 2] = \begin{bmatrix}0&0\\0&4\end{bmatrix}$ $\quad \begin{bmatrix}1\\1\end{bmatrix}[1\ 1] = \begin{bmatrix}1&1\\1&1\end{bmatrix}$

$\begin{bmatrix}0\\3\end{bmatrix}[0\ 3] = \begin{bmatrix}0&0\\0&9\end{bmatrix}$ $\quad \begin{bmatrix}1\\2\end{bmatrix}[1\ 2] = \begin{bmatrix}1&2\\2&4\end{bmatrix}$

d) for the correlation matrix:

if $\lambda_1 \cdot \lambda_2 > \tau$, we detect a corner

where λ_1 & λ_2 are the largest eigen value of A

& τ is a threshold value.

e) For corner detection, non-maximum supression works as:

- Compute $\lambda_1 \cdot \lambda_2$ for all locations.
- sort pixels based on $\lambda_1 \cdot \lambda_2$, in descending order.
- Start from the top, select a strongest corner
- delete corners in vicinity of the selected corner.
- Stop when we have detected X% of the pixels to ~~be cornersm~~ have corners.

f) Harris Corner detection algorithm avoids computing the eigenvalues of the gradient correlation matrix directly by computing determinant and trace of the gradient matrix as $\det(C)$ is same as $\lambda_1 \cdot \lambda_2$ and trace(C) is same as $\lambda_1 + \lambda_2$

g) For better localization of a corner, we try to find the best hypothesis P, by projecting gradients onto edge hypothesis and choose P with minimal projection.

Where the objective function is $E(p) = \sum_{i=1}^{n} ((p_i - p) \cdot \nabla I(p_i))^2$

i.e, projection of $\nabla I(p_i)$ onto $(p_i - p)$

$$p^* = \underset{p}{\operatorname{argmin}}\, E(p), \text{ so we solve } \nabla E(p) = 0.$$

$$\nabla E(p) = 2 \sum_{i=1}^{n} \nabla I(p_i) \nabla I(p_i)^T (p_i - p)$$

$$\Rightarrow \underbrace{\sum_i \nabla I(p_i) \nabla I(p_i)^T p_i}_{V} = \underbrace{\sum_i \nabla I(p_i) \nabla I(p_i)^T}_{C} p$$

$$\Rightarrow V = CP$$

$$P = C^{-1} V$$

for the solution to exist C^{-1} should exists.

C^{-1} exists as C is singular matrix & λ_1 & λ_2 are large as $\lambda_1 \cdot \lambda_2 > \tau$, where λ_1 & λ_2 are eigen values of C

h) Feature points can be characterized by HOG (Histogram of Oriented Gradients) by below steps:

1. take window
2. Split into blocks, (could be overlapping)
3. Compute histogram of gradient orientation in local blocks.
4. Concatenate histograms.

Requirements from a good characterization of feature points.

1. feature point should be translation invariant
2. — " — Rotation. "
3. — " — Scale "
4. — " — illumination "

i) SIFT features are computed by.

- taking large neighborhood of pixels
- break the neighborhood into smaller blocks.
- gradient vector is computed for each ~~block~~ pixel in the block.
- orientation histogram is computed for each block. by accumulating the gradients of the pixels
- finally the histograms are concatenated to compute the features.

2 a) the problem of using slope and y-intercept as line parameters with Hough transform is that the slope can get to infinity as the line is parallel to y axis. And also the y-intercept can range from $[-\infty, \infty]$. So, we cannot compute all parameters and know in prior how much space to reserve to store these parameters.

b) Slope of 45°
distance of 10 from origin

~~a = cos 45° = 0.5125~~
~~b = sin 45° = 0.851~~
~~d = 10~~

$d = x \cos\theta + y \sin\theta$ $\quad \cos 45° = \frac{\sqrt{2}}{2}$

$10 = x \cos 45° + y \sin 45°$ $\quad \sin 45° = \frac{\sqrt{2}}{2}$

~~10 = 0.5 20x + 0.851 y~~

$10 = \frac{\sqrt{2}}{2}(x+y) \Rightarrow x+y = \frac{20}{\sqrt{2}} \Rightarrow x+y = 14.14$

$\Rightarrow a = 1,\ b = 1,\ c = 14.14$

c) When using the polar representation of lines, each point in image votes with a sinusoidal curve in the parameter space.

d) lines are detected in the parameter plane by taking the ~~votes~~ point where all the votes intersect. This point defines the parameters of the line a, b & c. Using these parameters we can construct the line as $ax + by + c = 0$.

e) trade-off with bin size in parameter plane:

- Big bin size:
  - we get less votes & may miss out some information
  - less accurate as we miss out some information.
  - less computation to do.
  - less sensitive to noise.
- Small bin size:
  - we get more votes & may be susceptible to noise
  - more computation to do.
  - votes may not intersect as we have more accurate information

f) Voting in the parameter plane can be improved if the normal at each voting point is known as instead of using the range of θ from 0 to 180 degree we can just scan over over range θ_{min} to θ_{max}.

This can be achieved by finding θ ~~from~~ using the ∇I at each voting point and scan over $(\theta - \nabla I, \theta + \nabla I)$, which is computationally more efficient.

g) dimension of parameter space is n when using Hough transform for circle. $f((x, y), a_1, a_2, \ldots, a_n)$

to get a vote we scan over $a_1, a_2 \ldots a_{n-1}$ & find a_n for each (x, y)

3 a) equation $y = ax + b$ for line fitting only minimize the algebraic distance of the actual points to the predicted points on the line i.e, it doesn't lead to an optimal solution.

Lines whose slope is bigger cannot fit accurately with this equation.

b) normal : $(1, 2)$

distance from origin : 2

$l = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ ~~[illegible]~~ $l = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$

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

c) using explicit line equation $x\cos\theta + y\sin\theta = d$, given the slope θ and distance from origin d, we find the line coefficients a, b, c.

We have to solve $Cl = 0$, where C is a 3x3 matrix as written above, to get the unknown line parameters.

$$\left(C = \sum_{i=1}^{n} x_i x_i^T \right) \text{ or } C = D^T D \text{ where } D = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_n & y_n & 1 \end{bmatrix}$$

e) $P_i = (x_i^2, x_i y_i, y_i^2, x_i, y_i, 1)$

and $S = \sum_{i=1}^{n} P_i P_i^T$

are the explicit equatiom of the Conic curvee.

$ax^2 + bxy + cy^2 + dx + ey + f = 0$ is the implicit equation

$b^2 - 4ac < 0$, guarantees that the model will be an ellipse.

f) Given $\{x_i, y_i\}_{i=1}^{n}$

we have to solve $E(l) = \sum_{i=1}^{n} (l^T P_i)^2$, where $P_i = (x_i^2, x_i y_i, y_i^2, x_i, y_i, 1)$,

to ~~solve~~ fit an ellipse using algebraic distance.

$Sl = 0$, where $S = \sum_{i=1}^{n} P_i P_i^T$

give the solution. solution is eigenvector corresponding to 0 eigen value.

Points closer to the short axis of the elipse have more effect on the fitlinge as these points get more weight considering the algebraic distance of these are lesser than those closer to the long axis of the ellipse.

g) objective function to be minimized when fitting an ellipse using geometric distance: $E(l) = \sum_i \frac{|f(P_i;l)|}{|\nabla f(P_i;l)|}$ where $|f(P_i;l)| = (l^T P_i)^2$ & $P_i = (x_i^2, x_i y_i, y_i^2, x_i, y_i, 1)$

Complication involved here is that, this doesn't result in a Quadratic equation, so we don't get an explicit solution and a linear solver can not be used. so, we have to do an iterative approach like gradient descent.

h) Objective function of active contours:

$$E[\phi(s)] = \int_{\phi(s)} \left(\alpha(s)\, E_{continuity} + \beta(s)\, E_{curvature} + \gamma(s)\, E_{img}\right) ds$$

Where $\alpha(s)$, $\beta(s)$ & $\gamma(s)$ are coefficients (variable)

$E_{continuity}$, $E_{curvature}$ & E_{img} are energy terms.

$\alpha(s) E_{continuity} + \beta(s) E_{curvature}$ is ~~called~~ Internal part

and $\gamma(s) E_{img}$ is the external part

We want $E_{continuity}$ & $E_{curvature}$ to be smaller

and E_{img} to be high i.e, high integrated gradients.

and $E_{continuity} = \left|\frac{\partial \phi}{\partial s}\right|^2$, $E_{curvature} = \left|\frac{\partial^2 \phi}{\partial s^2}\right|^2$ and $E_{img} = -|\nabla I|^2$

i) when the curve is discrete & we use active contours.

$E_{continuity}$ is estimated as distance between neighboring points.

i.e, $E_{continuity} = \sum_i |P_i - P_{i-1}|^2$

$E_{curvature}$ is estimated as difference of tangents at neighboring points

$$E_{curvature} = \sum |(P_{i+1} - P_i) - (P_i - P_{i-1})|^2$$

$$= \sum |P_{i+1} - 2P_i + P_{i-1}|^2$$

j) the continuity of active contours may be relaxed or to allow discontinuity, we find high curvature points and set $\beta_i = 0$ i.e, if $|P_{i+1} - 2P_i + P_{i-1}| > \tau$

then $\beta_i = 0$.