Assignment_2 Solution

1. Corner Detection

a.

1. Check the local neighborhood to see if there are more than one direction or orientation of gradient.
2. The number of principal directions is derived as follows:
   1. Find correlation matrix in local neighborhood
   2. Find eigenvalues of matrix
   3. Check that $\lambda_1 \cdot \lambda_2 > \tau$

b.

Given $\{p_i\}_{i=1}^n$, find v to minimize projection of points p_i in v:

$$E(v) = \sum_{i=1}^{n} (p_i v)^2 = v^T \left(\sum_{i=1}^{n} p_i p_i^T \right) v$$
$$\nabla E(v) = 0 \Rightarrow Av = 0$$

The solution is the eigenvector of matrix A belonging to zero eigenvalue.

c. $\begin{bmatrix} 4 & 6 \\ 6 & 44 \end{bmatrix}$

d. $\lambda_1 \cdot \lambda_2 > \tau$, where λ_1 and λ_2 are the eigenvalue of gradient correlation matrix, and τ is the threshold

e.

- Get λ_1 and λ_2 for each point
- Compute $\lambda_1 \cdot \lambda_2$ and sort in decreasing order
- Select top points as corner and delete the neighboring points
- Stop deleting corner candidates after a desired percentage of the corners have been detected

f. $C(G) = \det(G) - k \cdot tr^2(G)$, where $\det(G)$ is $\lambda_1 \cdot \lambda_2$, and $tr^2(G)$ is $\lambda_1 + \lambda_2$

g.

1. $P = C^{-1}v$, the condition is $\lambda_1 \cdot \lambda_2 > \tau$ in C.
2. C must be a non-singular matrix

h.

1. Point characterized using HOG:

    - Take window and split into blocks
    - Compute histogram of gradient orientations
    - Concatenate histogram
2. A good characterization of feature points should be translation, rotation, scale, and illumination invariant.

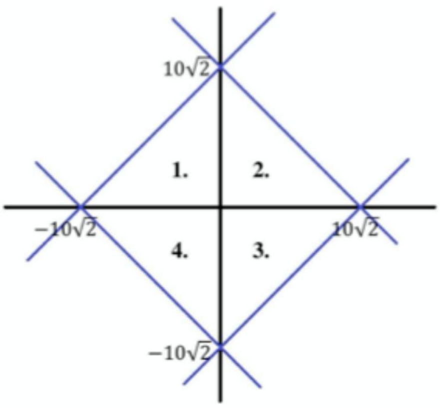
i. First a set of orientation histograms is created on 4×4 pixel neighborhoods with 8 bins each. These histograms are computed from magnitude and orientation values of samples in a 16×16 region around the key point such that each histogram contains samples from a 4×4 subregion of the original neighborhood region. The magnitudes are further weighted by a Gaussian function with σ equal to one half the width of descriptor window. The descriptor then becomes a vector of all the values of these histograms. Since there are $4 \times 4 = 16$ histograms each with 8 bins the vector has 128 elements.

---

2. Line Detection

a. The range of slope and y-intercept are infinity

b. Multiple solutions:

<table>
<tr><th>Equation</th><th>Graph</th></tr>
<tr><td>$a = -b, c = 10\sqrt{2}a$
$a = b, c = -10\sqrt{2}a$
$a = -b, c = 10\sqrt{2}b$
$a = b, c = 10\sqrt{2}b$</td><td>
</td></tr>
</table>

c. Line appears as sinusoid curve

d. Each edge point is transformed to a line in the Hough space, and the areas where most lines intersect in the Hough space is interpreted as true lines in the edge map.

e.

- Big bin: fewer votes, faster, lower accuracy
- Small bin: more votes, slower, higher accuracy

f. Instead of $\theta \in [0, 180] \rightarrow \theta \in [\theta - \Delta\theta, \theta + \Delta\theta]$

g. 3 dimensions: a, b, r

---

3. Model Fitting 1

a.

1. With $y = ax + b$, we cannot model vertical or near vertical lines because the slope dy/dx would have to be infinite. This shows that we need to be careful when choosing the model so that it can describe all possible (and not only a subset) of observations.
2. Vertical and near vertical lines.

b. $[1/\sqrt{5}, 2/\sqrt{5}, -2]^T$

c.

Explicit Equation:

1. Explicit line equation: $y = ax + b$

   $E(a, b) = \sum_{i=1}^{m} (y_i - (ax_i + b))^2$

   $a^*, b^* = \arg\min_{a,b} E(a, b)$

   $\Rightarrow \nabla E(a, b) = 0 \Rightarrow a^*, b^*$

2. The equation need to solve:

   $x = A^{-1}b$ where $A = \begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & m \end{bmatrix}$ and $b = \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix}$

Implicit Equation:

1. Implicit line equation: $l^T x = 0$

2. The equation need to solve:

   $E(l) = l^T Sl \to \nabla E(l) = 0 \to Sl = 0$, where $l = \begin{bmatrix} \sum x_i^2 & \Sigma x_i y_i & \Sigma x_i \\ \sum x_i y_i & \Sigma y_i^2 & \Sigma y_i \\ \sum x_i & \sum y_i & n \end{bmatrix}$

d. $\begin{bmatrix} 5 & 15 & 3 \\ 15 & 46 & 10 \\ 3 & 10 & 3 \end{bmatrix}$

e.

1. Implicit equation: $ax^2 + bxy + cy^2 + dx + ey + f = 0$
2. Constraint: $b^2 - 4ac < 0$

f.

1. - $E(l) = \sum_{i=1}^{n} (l^T p_i)^2 + \lambda(l^T Cl + 1) = l^T Sl + \lambda(l^T Cl + 1)$
   - Solve $\nabla E(l) = 2Sl + \lambda 2Cl = 0 \to l = \lambda S^{-1} Cl$

- The solution is the eigenvector of $S^{-1}C$ belonging to its negative eigenvalue

2. Short axis:

   algebraic distance: $q_i \equiv l^T p_i \approx \frac{d_i}{d_i+r_i} \Rightarrow \frac{d}{d+r_1} > \frac{d}{d+r_2}$, where r_1 is the short axis and r_2 is the long axis

g.

1. $E(l) = \sum_i \frac{|f(P_i;l)|}{|\nabla f(P_i;l)|}$
2. No explicit solution to $\nabla E(l) = 0 \Rightarrow$ iterative numerical solution (gradient decent)

h.

$$E(\varphi(s)) = \int (\alpha E_{\text{continuity}} + \beta E_{\text{curvature}} + \gamma E_{\text{image}})\, ds$$

$$\text{Where } E_{\text{continuity}} = \left|\frac{\partial\varphi}{\partial s}\right|^2, \quad E_{\text{curvature}} = \left|\frac{\partial^2\varphi}{\partial s^2}\right|^2, \quad E_{\text{image}} = \Sigma - |\nabla I\,(P_i)|^2$$

i.

- $E_{\text{continuity}} = \left|\frac{\partial\varphi}{\partial s}\right|^2 = \sum |P_i - P_{i-1}|^2$
- $E_{\text{curvature}} = \left|\frac{\partial^2\varphi}{\partial s^2}\right|^2 = \sum |(P_{i+1} - P_i) - (P_i - P_{i-1})|^2 = \sum |P_{i+1} - 2P_i - P_{i-1}|^2$

j. In high curvature point, we set $\beta_i = 0$ in order to accommodate for sharp corners:

If $|P_{i+1} - 2P_i - P_{i-1}| > \tau \rightarrow \beta_i = 0$

---

4. Model Fitting 2

a. $\begin{bmatrix} 10 & 14 \\ 14 & 20 \end{bmatrix}$

b.

1. $-x + y = 0$
2. Normalized normal: $[-1/\sqrt{2}, 1/\sqrt{2}]$

c. -2.5

d. 1

e. $\begin{bmatrix} 10 & 14 & 4 \\ 14 & 20 & 6 \\ 4 & 6 & 2 \end{bmatrix}$

f. $1/2$

g.

- Approximated geometric distance: $1/2$
- Algebraic distance: 1

h.

- $E_{\text{continuity}} = 2$
- $E_{\text{curvature}} = 0$

i. $\beta = 0$

---

5. Robust Estimation

a.

1. Outliers are the observation point that is distant from other observations.
2. The fundamental problem is that the model fitting can be mismatched by the influence of outlier.

b.

1. $E(\theta) = \sum_{i=1}^{n} \xi_\sigma(d(x_i;\theta))$
2. In robust estimation, the function give low weight for high value outlier, however the least squares objective function give higher weight for high value outlier.

c.

1. $\xi_\sigma(x) = x^2/(x^2+\sigma^2)$
2. Advantage:
   - we can control the loss function
   - the weight of outliers is up to 1 (has upper bound)
3. Start with large σ and decrease as converging: $\sigma^{(n)} = 1.5 \times \text{median}\{d(x_i;\theta^{(n-1)})\}$

d. $1/2$

e.

1. The RANSAC algorithm is a learning technique to estimate parameters of a model by random sampling of observed data. Given a dataset whose data elements contain both inliers and outliers, RANSAC uses the voting scheme to find the optimal fitting result.
2. It should be small, because it can avoid include more outliers.

f.

1. Parameter of RANSAC algorithm:
   - n: is the number of points at each evaluation
   - d: is the minimum number of points needed
   - t: is the threshold to identify inliers
   - k: is the number of trials

2. $k = \log(1-p)/\log(1-w^n)$

g. $k = \log(0.01)/\log(1-0.9^n)$

---

6. Segmentation and Recognition

a. The objective of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.

b.

1. The merge approach starts with each pixel as a separate cluster and iterates merging clusters,
2. The split approach starts with all pixels as one cluster and iterating splitting the cluster.

c. k-means is to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

- Select k (number of clusters)
- Select initial guess of mean: $\mu_1, \mu_2, \ldots, \mu_k$
- Repeat while μ_j change:
  - $l_i = \arg\min_{j \in [1,k]} \|f_i - m_j\|^2$
  - $s_j = \{i | l_i = j\}$
  - $m_j = \sum_{i \in s_j} f_i / |s_j|$

d.

1. Difference between graph cut and normalized cut:
   - In graph cut, the weight is computed as: $w_{pq} = \exp(-\|f(p) - f(q\|))$ and we remove weak links to generate segmentations.
   - In normalized cut, instead of just using the weight, we add cut cost to weights then we normalize the it by taking account the size of produced clusters.
2. To avoid generating single node

e.

- Similarity matrix: $W = \begin{bmatrix} w_{1,1} & \cdots & w_{1,100} \\ \vdots & \ddots & \vdots \\ w_{100,1} & \cdots & w_{100,100} \end{bmatrix}_{100\times100}$
- Weighted degree matrix: $D = \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_{100} \end{bmatrix}_{100\times100}$ where $d_i = \sum_{j=1}^{100} w_{i,j}$
- Laplacian matrix: $L = D - W$

f. Cut is a sequence of number $+1$ or -1 ($X = \{1, -1\}^n$) to present two group we created after cut. If the $X(i) = 1$, the node i belonging to cluster A. If $X(i) = -1$, the node i belonging to cluster B.

$$
\begin{aligned}
N_{\text{cut}}(A,B) &= \frac{Cut(A,B)}{Vol(A)} + \frac{Cut(A,B)}{Vol(B)} \\
&= \frac{\sum_{(x_i>0,x_j<0)}(-W_{i,j}x_ix_j)}{\sum_{x_i>0} d_i} + \frac{\sum_{(x_i>0,x_j<0)}(-W_{i,j}x_ix_j)}{\sum_{x_j<0} d_j}
\end{aligned}
$$

g.

$$
\begin{aligned}
&\min_x N_{\text{cut}}(x) = \min_y \frac{y^T(D-W)y}{y^T Dy} \\
&\text{s.t.} \quad y^T D\mathbf{1} = 0 \quad \text{where} \quad \mathbf{1} \equiv [1,\ldots,1]
\end{aligned}
$$

h.

$$
\begin{aligned}
&\min_x N_{\text{cut}}(x) = \min_y \frac{y^T(D-W)y}{y^T Dy} \\
&\text{s.t.} \quad y^T D\mathbf{1} = 0 \quad \text{where} \quad \mathbf{1} \equiv [1,\ldots,1]
\end{aligned}
$$

Solution $(D-W)y = \lambda Dy \Rightarrow$ The eigenvector of the second smallest eigenvalue is the solution

i.

1. Eigenfaces approach:
    - map image to lower dimensional vector using PCA
    - measure similarity to templates in lower dimensional space
2. Limitation:
    - In low dimensional space, it is less sensitive to small variations

j.

1. bag-of-words approach:
    - Extract features (e.g. SIFT or HOG)
    - Cluster features to create a codebook
    - Compute a distribution of code words in each class
    - Classify using distribution of code word
2. Limitation:
    - The classifier looks different local neighborhoods to make decision but not look the relationship between them.