QUESTION 1
HOUGH TRANSFORM

- Transform a point in a cartesian coordinate to
- all lines (line parameters) that pass the point in the polar coordinate

\[ H : (x, y) \rightarrow \{ (r, \theta) | r(\theta) = x \cos \theta + y \sin \theta \} \]

\[
y = -\frac{\cos \theta}{\sin \theta} x + \frac{r}{\sin \theta} \rightarrow r(\theta) = x \cos \theta + y \sin \theta
\]
feature $\Phi(x)$ to points in the 2D Hough space $S = \{\vec{a}\}$
- $\vec{a}$ is the reference origin of the shape
- For points $\vec{x}$ on the boundary, store $\vec{r} = \vec{a} - \vec{x}$ : R-Table
  - As a function of the gradient $\Phi(\vec{x})$
  - Rotation and scale : 4D Hough Space

IMPLICIT SHAPE MODEL

- R-Table as a function of an image patch
- transform an image patch to object centers

B. Leibe, A. Leonardis, and B. Schiele, Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV 2004
QUESTION 2
BAG OF VISUAL WORDS

Feature Extraction

Clustering

Encoding

Spatial Binning

Kernel SVM

\[ k(x, y) = \langle \Psi(x), \Psi(y) \rangle \]
BAG OF VISUAL WORDS

- Feature Extraction
  - Local image feature
  - Robust to typical image transformations
    - Dense SIFT
    - SIFT at every location vs a key point
    - Interest points might not correspond to foreground

- Clustering (Dictionary)
  - Visual words should be distinctive and diverse
  - Common visual words (wheel) will form a cluster
    - K-means clustering
**Bag of Visual Words**

- **Encoding**
  - Maps local features to visual words
  - Hard quantization
    - assign to the NN

- **Spatial Pyramid**
  - Encode spatial information
  - Divide the image into subsections
  - For each subsection, create a VW histogram
SIMILARITY METRIC FOR DISTRIBUTIONS

Train a linear SVM on features (distribution): Distance metric??

How similar is a distribution $P$ from a distribution $Q$: $f$-divergence!

$f : (0, \infty) \rightarrow \mathcal{R}$, convex

$$D_f(P \parallel Q) = \int f \left( \frac{dP}{dQ} \right) dQ$$

- Kullback-Leibler Divergence
  $$f(x) = x \log x$$
- $\chi^2$ Divergence
  $$f(x) = (x - 1)^2$$
FOR A DISCRETE CASE

- Intersection kernel: \( k(x, y) = \sum \min(x_i, y_i) \)
- \( \chi^2 \) kernel: \( k(x, y) = \frac{1}{2} \sum \frac{(x_i - y_i)^2}{x_i + y_i} \)

In the problem 2, we will use \( \chi^2 \) kernel for the similarity metric
HOMOGENEOUS KERNEL AND FINITE APPROXIMATION

$K(x, y)$ expensive, non linear!
But in the feature space $\Psi(\cdot)$ it becomes linear ($\infty$ dimension)

- Feature map $\Psi(x)$ using the Homogeneous Kernel
- Generic histogram kernel is $K(x, y) = \sum_i k(x_i, y_i)$
  Homogeneous if $k(cx, cy) = ck(x, y)$
  - intersection, $\chi^2$, Hellinger's, Jensen-Shannon's
- Find a finite feature map $\hat{\Psi}(\cdot)$ that
  
  $$k(x, y) = \langle \Psi(x), \Psi(y) \rangle \approx \langle \hat{\Psi}(x), \hat{\Psi}(y) \rangle$$

- Map distributions using $\hat{\Psi}(\cdot)$ then it becomes a linear classification problem in the feature space!
**HOMOGENEOUS KERNEL**

- Use the homogeneity, 
  \[ k(x, y) = \sqrt{xy} k\left(\frac{x}{y}, \frac{y}{x}\right) = \sqrt{xy} K(\log(y/x)) \]
- Ex, intersection kernel
  - \[ k(x, y) = \min(x, y) = \sqrt{xy} K(\log \frac{y}{x}) \]
  - \[ K(w) = e^{-|w|/2} \]
- Define \( \kappa(\lambda) \) as the Fourier transformation of \( K(w) \)
- \[ \Psi(x) = e^{-i\lambda \log x} \sqrt{xk(\lambda)} \]
  - Continuous, infinite dimensional feature
- Sample at points \( \lambda = -nL, (-n + 1)L, \ldots, nL \)
- \( \hat{\Psi}(x) \in \mathcal{R}^{2n+1} \)
  - \[ k(x, y) \approx \langle \hat{\Psi}(x), \hat{\Psi}(y) \rangle \]
- In the problem 2, we use \( n = 1 \)
- vl_homkermap()
BAG OF VISUAL WORDS

Feature Extraction

Clustering

Encoding

Spatial Binning

Kernel SVM

\[ k(x, y) = \langle \Psi(x), \Psi(y) \rangle \]
BAG OF VISUAL WORDS

- Install VL Feat, an extensive CV library
  - [http://www.vlfeat.org/](http://www.vlfeat.org/)
  - Set up the path correctly in the starter code `p2.m`
- Vary options and see how it affects the performance
• **Code** `extract_dense_sift.m`
  - Image feature extraction
    - Dense SIFT features
    - `use vl_dsift()`
  - randomly select 10,000 features

• **Code** `create_dictionary.m`
  - Create dictionary of visual words
    - Use k-means to find the centroid of clusters.
    - `use vl_kmeans()`

• **Code** `create_histograms.m`
  - Represent images as histograms
  - Spatial pyramid