iMorpheus.ai

Semantic-Preserving BoW Models and Applications

Learning BoW by Y. Li @ iMorpheus.ai

Date Jan. 12, 2018 12:00PM

Journal Club

Journal club介绍与自动驾驶中定位方案相关的论文， 主要关注的方向有：SLAM算法、点云数据的处理和压缩、 特征地图、传感器数据处理和融合、 GNSS信号处理等。我们一直关注领域前沿技术，选取得到广泛认可的、或者是在我们的实际使用中结果比较好的论文， 与大家分享，共同学习成长。

每周五 北京时间12点

http://imorpheus.ai/journalclub

扫码加入无人驾驶技术群

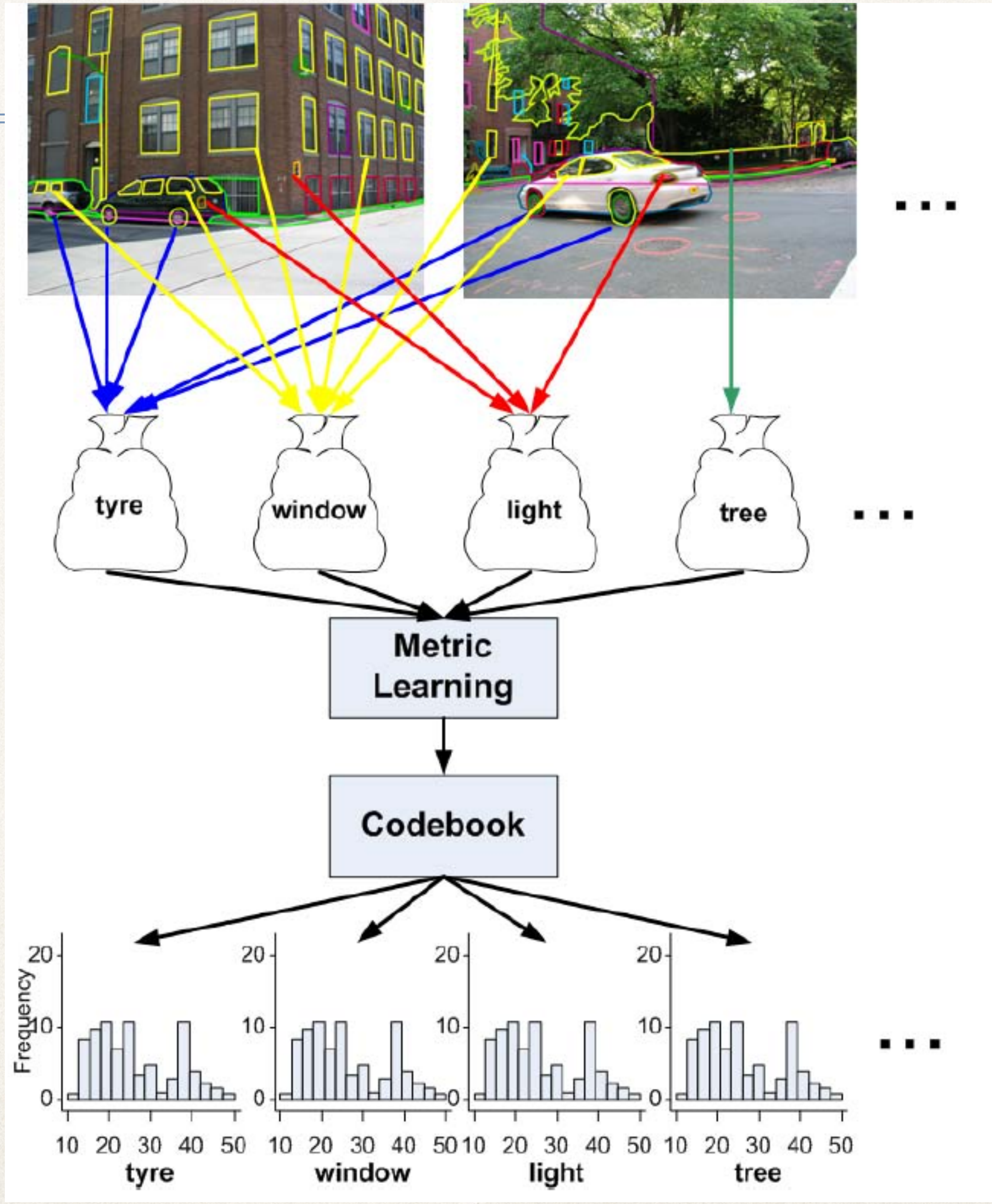
iMorpheus.ai

Background

- BoW deals with an image:
  - Interest point detector (e.g. Difference of Gaussians) to detect salient patches/regions in the image.
  - Feature descriptor (e.g. SIFT) to represent the local patches/regions as numerical feature vectors.
  - To generate a codebook by converting the patches to "codewords", e.g. applying k-means clustering and defining codewords based on the centers of the clusters.
- Drawbacks of BoW:
  - Ignorance of spatial information
  - Semantics of objects is considerably lost

iMorpheus.ai

Semantic Gap ->a distance metric learning method



Data descriptions

- Training data:
  - MIT label testbed of images（495 objects， 185 images， 400000 features），VOC2006
- Side information:
  - considering pairwise feature instance *xi1, xi2,*
  - zi1 and zi2 are binary indicators to indicate whether a feature instance is located at the object region or the background region in the image.
  - yi shows if both xi1 and xi2 are on the same semantic parts of objects

- The basic Principle of metric learning:

$$d(x_{i1}, x_{i2}) = \sqrt{(x_{i1} - x_{i2})^\top A (x_{i1} - x_{i2})} \qquad (1)$$

A. The distances between visual feature vectors of the same semantics should be minimized.

B. Distances between feature vectors of different semantics should be maximized.

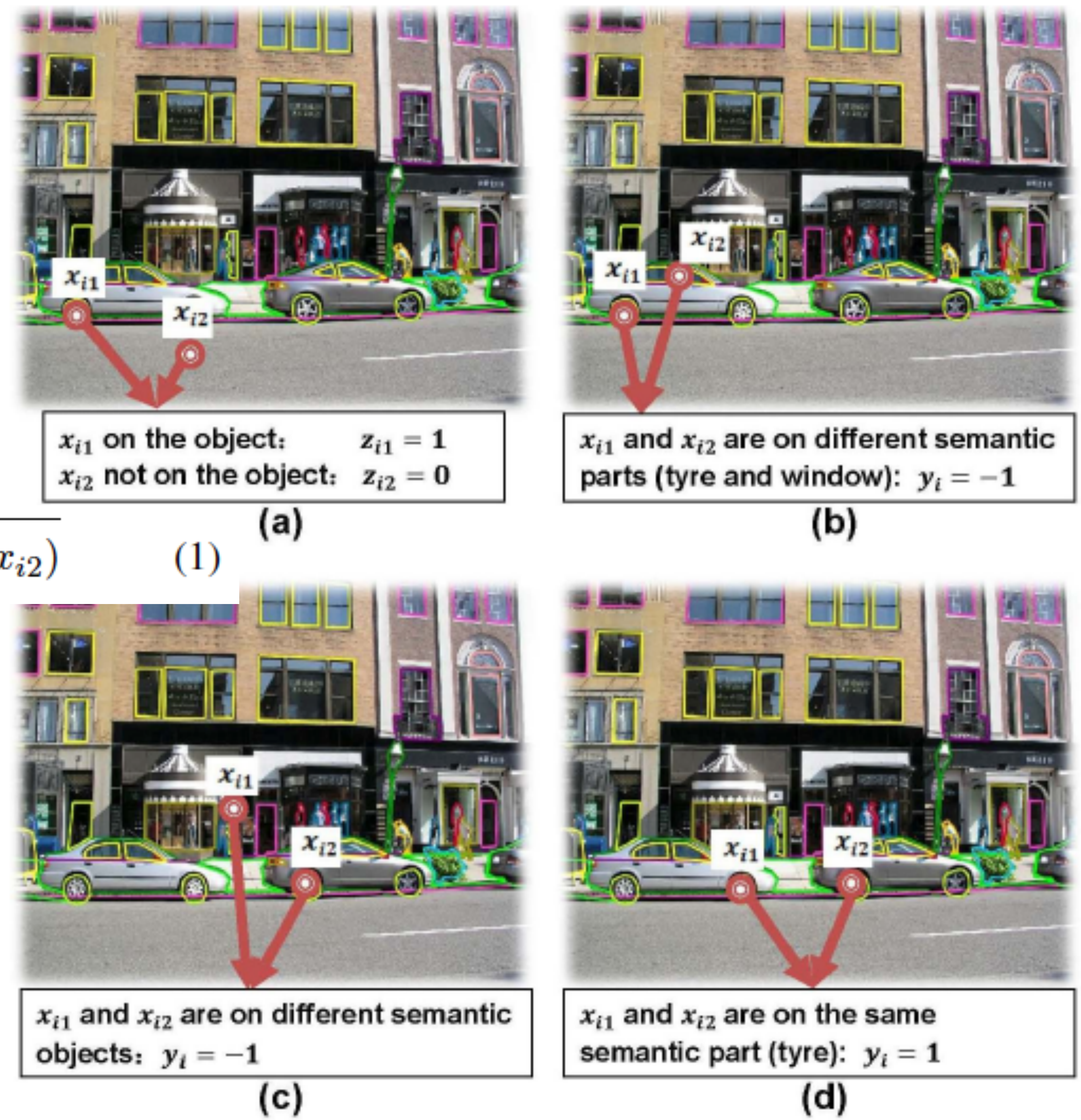


Fig. 2. Illustration of side information between objects and feature instances.

1. The distance metric estimation

$$\min_{A \succeq 0, b} \quad \sum_i z_{i1} z_{i2} \xi_i + \frac{\lambda}{2} tr(AA^\top) \qquad (2)$$

$$s.t. \quad y_i(\|x_{i1} - x_{i2}\|_A - b) \leq \xi_i, \xi_i \geq 0, i = 1, \ldots, n \qquad (3)$$

$$\|A\| = 1/\sqrt{\lambda} \qquad (4)$$

- Mahanalobis distance between two features under metric A
- Matrix:

$$X_1 = [x_{11}, x_{21}, \cdots, x_{n1}]^\top \quad X_2 = [x_{12}, x_{22}, \cdots, x_{n2}]^\top$$

$$Z_1 = \mathrm{diag}(z_{11}, z_{21}, \cdots, z_{n1}) \quad Z_2 = \mathrm{diag}(z_{12}, z_{22}, \cdots, z_{n2})$$

$$Y = \mathrm{diag}[y_1, \cdots, y_n]$$

- A gradient search algorithm

iMorpheus.ai

Algorithm 1 The Semantics-Preserving Metric Learning (SPML) algorithm

INPUT:

- SIFT feature matrix: $X \in \mathbb{R}^{N \times d}$
- pairwise constraint $(x_{i1}, x_{i2}, z_{i1}, z_{i2}, y_i)$, where x_{i1} is the i_1^{th} SIFT feature, z_{i1} indicate whether the location of feature x_{i1} is on the semantic object, and constraints $y_i = \{+1, 0, -1\}$ represents feature x_{i1} and x_{i2} are on the same semantic part of the object, not known, or on different semantic parts.
- regularization parameter λ
- learning rate parameter γ

PROCEDURE:

1: initialize metric and threshold: $A = I$, $b = b_0$

2: set iteration step t = 1;

3: **repeat**

4: (1) update the learning rate:
$\gamma = \gamma / t$, t = t + 1

5: (2) update the subset of training instances:
$\mathcal{S}_t^+ = \{(x_{i1}, x_{i2}, y_i) | (1 + y_i)\|x_{i1} - x_{i2}\|_A^2 > 1\}$
$\mathcal{S}_t^- = \{(x_{i1}, x_{i2}, y_i) | (1 - y_i)\|x_{i1} - x_{i2}\|_A^2 < 1\}$
$\mathcal{S}_t = \mathcal{S}_t^+ \bigcup \mathcal{S}_t^-$

6: (3) compute the gradients w.r.t. A
$\nabla_A \mathcal{L} \leftarrow Z_1 Z_2 (\lambda A + D_X^\top Y^\top D_X)$,
$D_X = X_1 - X_2$,

7: (4) compute the gradients w.r.t. b
$\nabla_b \mathcal{L} \leftarrow \mathrm{tr}(Z_1 Z_2 Y)$

8: (5) update metric and threshold:
$A_{t+1} \leftarrow A_t - \frac{\gamma}{t} \nabla_A \mathcal{L}, \qquad b_{t+1} \leftarrow b_t - \frac{\gamma}{t} \nabla_b \mathcal{L}$

9: (6) project A back to the PSD cone:
$A_{t+1} = \sum_{i=1}^{d} \lambda_i \phi_i \phi_i^\top$
$A_{t+1} \leftarrow \sum_i \max(0, \lambda_i) \phi_i \phi_i^\top$

10: (7) normalize A_{t+1} to satisfy $\|A_{t+1}\| = \frac{1}{\sqrt{\lambda}}$:
$A_{t+1} \leftarrow \frac{1/\sqrt{\lambda}}{\|A_{t+1}\|} A_{t+1}$

11: **until** convergence

OUTPUT:

- feature metric A, threshold variable b

2. Codebook Generation

1. Codebook size assignment（Lmax=2500）

The number of codes -> visual complexity of an object category -> the diversity of its associated features

The generative probability of *xj* from the object *Ci* word-bag:

$$p(x_j|C_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{\|x_j - \hat{x}\|_A^2}{2\sigma^2}}$$

$$\hat{x} = \frac{1}{n_{C_i}} \sum_{x_j \in C_i} x_j$$

Information entropy:

$$H(C_i) = -\sum_{x_j \in C_i} p(x_j|C_i) \log p(x_j|C_i)$$

The number of codes or words:

$$L_{C_i} = \lfloor L_{max} \times \frac{H(C_i)}{\log n_{C_i}} \rfloor$$

2. Codebook generation

iMorpheus.ai

Algorithm 2 Codebook Generation Algorithm

INPUT:

- features and their object labels $\{(x, y), x \in \mathcal{X}, y \in \mathcal{C}\}$
- optimized distance metric A
- codebook size assigned for each object $L_{C_i}, i = 1, \cdots, M$
- the number of clusters for clustering $K > \max_i L_{C_i}$

PROCEDURE:

1: initialize the number of visual words $L = 0$
2: **for** $i = 1 : M$ **do**
3: clustering features of the i-th object $X_i = \{(x, C) | C = C_i\}$ into K clusters
$$[c_{ij}, r_{ij}] = kmeans(X_i, K)$$
4: calculate the size of each cluster:
$$S_{ij} = \sum_x \delta(\|x - c_{ij}\|_A, r_{ij})$$
5: sort clusters by their sizes
$$c_{ij} \leftarrow sort(c_{ij}, S_{ij}) \qquad r_{ij} \leftarrow sort(r_{ij}, S_{ij})$$
6: adopt top L_{C_i} largest clusters as visual words for the category
$$w_{L+j} = c_{ij}, r_{L+j} = r_{ij}, j = 1, \cdots, L_{C_i}$$
7: update the number of visual words $L = L + L_{C_i}$
8: **end for**

OUTPUT:

- the centers of visual words w_k and their range radius r_k, $k = 1, \cdots, L_{max}$

iMorpheus.ai

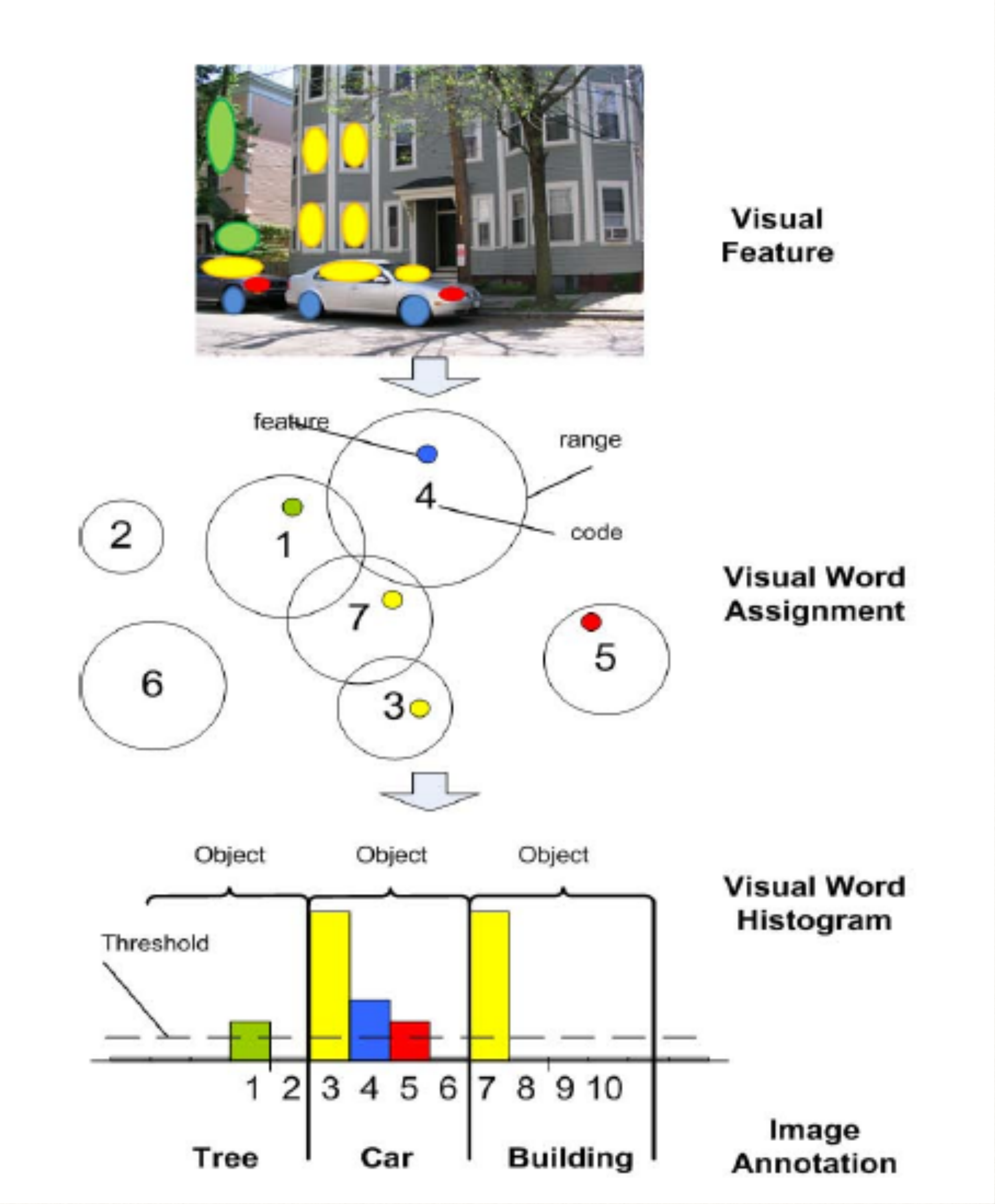
3. Visual word Histogram

- Difference from traditional BoW methods:
  - Traditional BoW assigns a feature to its closest visual words (a cluster).
  - In SPBoW, a feature can be assigned to multiple visual words in different objects.
- Visual word Histogram:

$$\pi(x,k) = \begin{cases} 1, & \|x - w_k\|_A < r_k; \\ 0, & \text{otherwise.} \end{cases}$$

$$f_I(k) = \textstyle\sum_{x \in I} \pi(x,k)$$

$$h_I(w_k) = \frac{f_I(k)}{\sum_{v=1}^{L_{max}} f_I(v)}$$



Application (Object Classification)

- Assuming that we are given a set of labeled image regions. Our goal is to automatically annotate a novel image *I*
  - Generative models

Probability of a visual word *wk* appearing in an object *Ci*, calculated based on the training set:

$$p(w_k|C_i) = \frac{\sum_{\{x|C(x)=C_i\}} \pi(x,k) + 1}{\sum_k \sum_{\{x|C(x)=C_i\}} \pi(x,k) + V}$$

The likelihood of object category Ci appearing in image I can be calculated by a Naive Bayes model :

$$p(C_i|I) \propto p(I|C_i)p(C_i) \propto p(C_i) \prod_k p(w_k|C_i)^{f_I(k)} \qquad (12)$$

The top N ranked categories are used to annotate the image

iMorpheus.ai

Application（Object Classification)

- Assuming that we are given a set of labeled image regions. Our goal is to automatically annotate a novel image *I.*
    - Discriminative models

We are given a set of training images (or image regions) and their semantic categories*{Ij, C(Ij)}j, j* is 1 to *Ntr*. Each image has visual word histogram $\mathbf{h}_I = [h_I(w_1), h_I(w_2), \cdots, h_I(w_{L_{max}})]$,Which is a Lmax-dimensional vector.

A multi-class classification task, based on the visual word histogram.

Using a binary SVM classifier to get the weight vector, w and b, for the *i*th category.

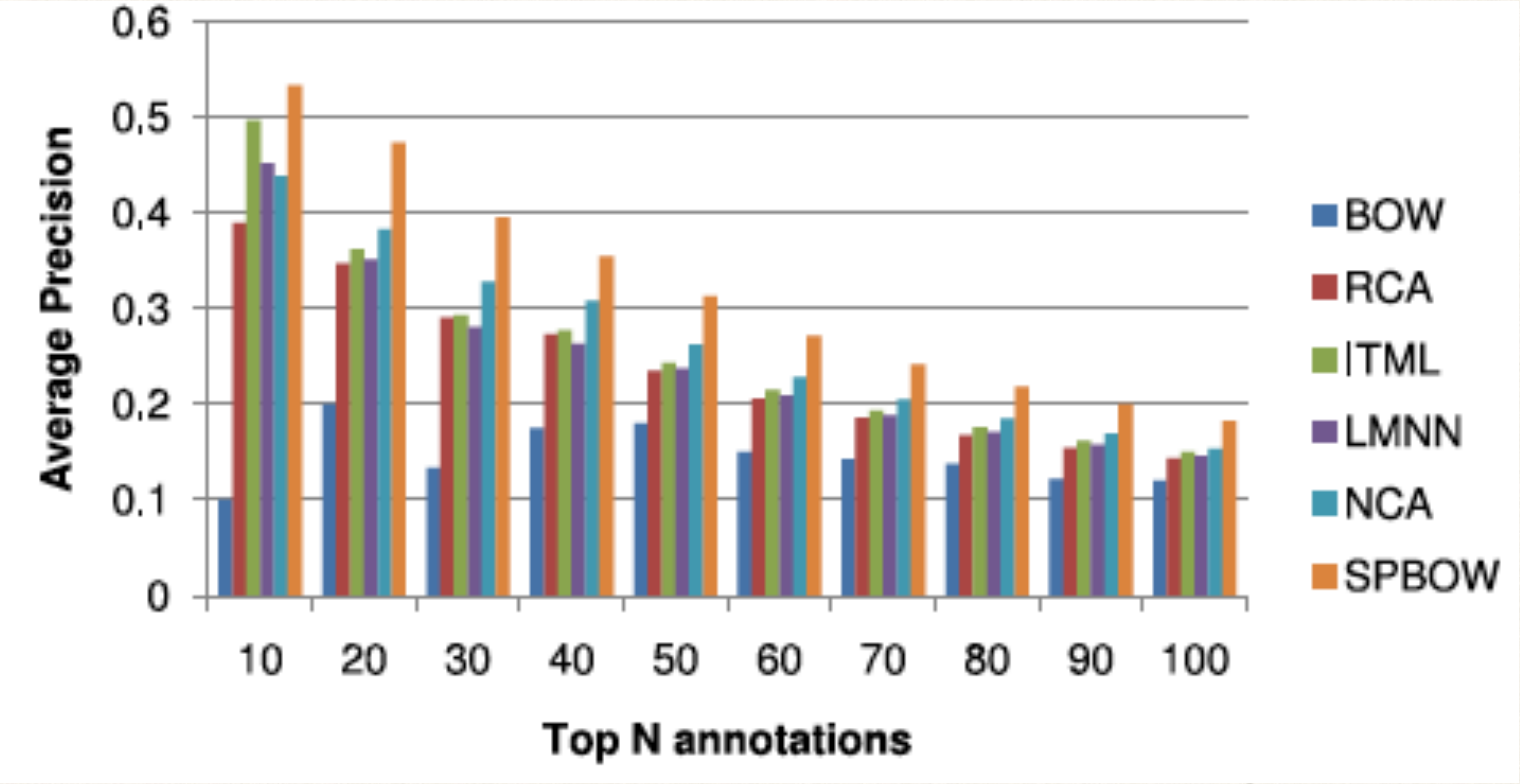
$$
\begin{aligned}
\min_{\omega,b} \quad & \frac{1}{2}\|\omega\|^2 + C\sum_j \xi_j \\
s.t. \quad & y_j(i)(\omega \cdot \mathbf{h}_{I_j} - b) \geq 1 - \xi_j, \xi \geq 0, 1 \leq j \leq N_{tr}
\end{aligned}
$$

A novel test image will be classified by all of the binary SVM classifiers, in which a positive output indicates that a specific object is detected on the image.

Results and Discussion

- Average Precision

  ~top N annotation



- Other influences: Number of constraints, codebook size and object categories (<10, Precision>90%)

- Codebook Generation:

Method	BoW	RCA	ITML	LMNN	NCA	SPBoW
Time Cost (s)	121	3	96	1759	457	8

iMorpheus.ai

- Thank You

Appendix

True Positive， False Positive，true negative，false negative

Precision其实就是在识别出来的图片中，True positives所占的比率：

P=Tp/（Tp+Fp）

其中的(True positives + False positives)也就是系统一共识别出来多少照片 。

在这一例子中，True positives为3，False positives为1，所以Precision值是 3/（3+1）=0.75。

意味着在识别出的结果中，飞机的图片占75%。

Recall 是被正确识别出来的飞机个数与测试集中所有飞机的个数的比值：

R=Tp/（Tp+Fn）

Recall的分母是(True positives + False negatives)，这两个值的和，可以理解为一共有多少张飞机的照片。

在这一例子中，True positives为3，False negatives为2，那么Recall值是 3/（3+2）=0.6。

意味着在所有的飞机图片中，60%的飞机被正确的识别成飞机.。

iMorpheus.ai

Journal Club

iMorpheus.ai Weekly Journal Club

Next Friday, 19/01/2018 12:00PM GMT+8

An introduction to Inertial Navigation

Website : **http://imorpheus.ai**
Email Address : **live@imorpheus.ai**

iMorpheus.ai