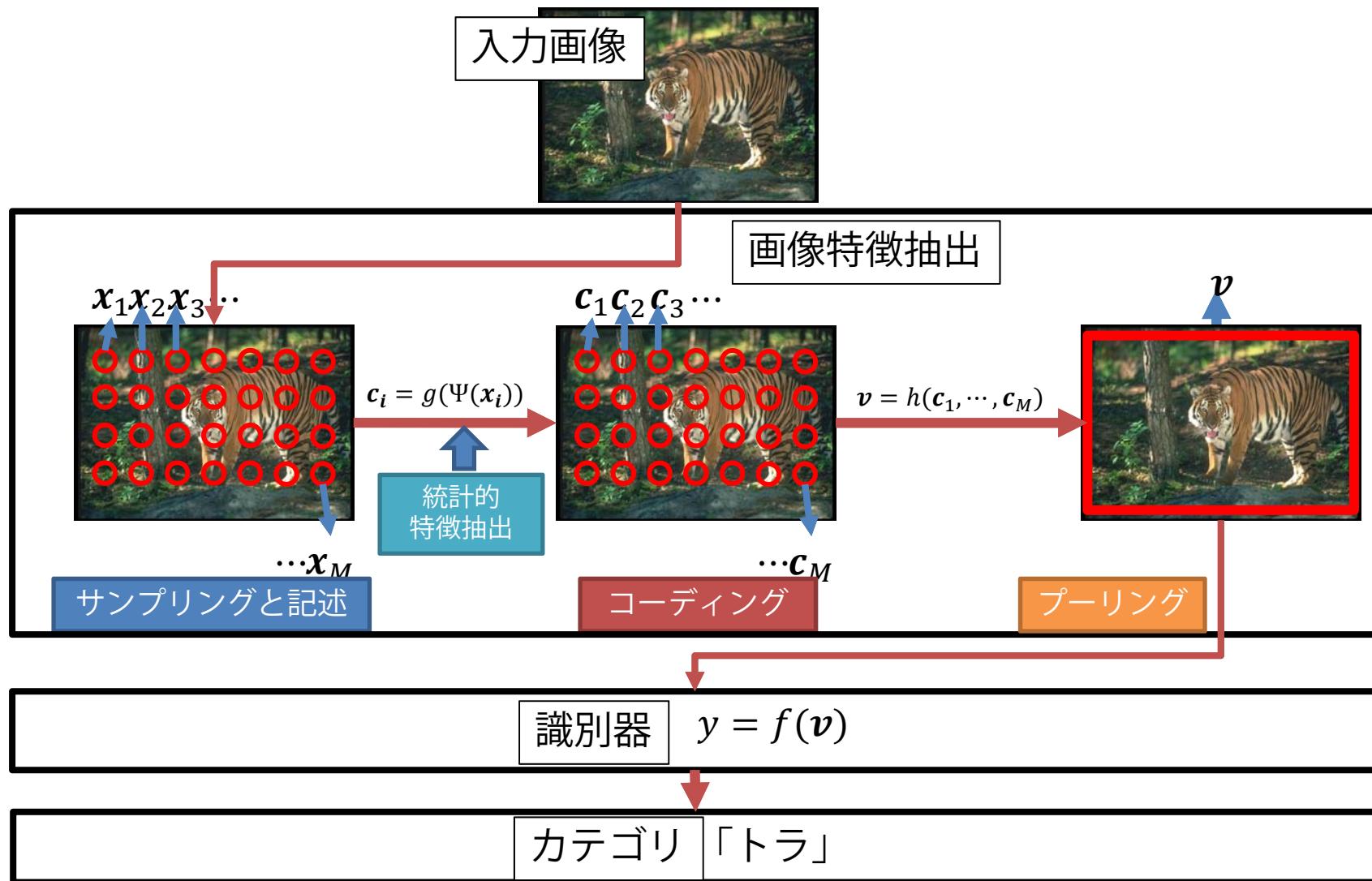


知能情報論 コーディング・プリント

2016年5月25日

東京大学 大学院情報理工学系研究科
原田達也

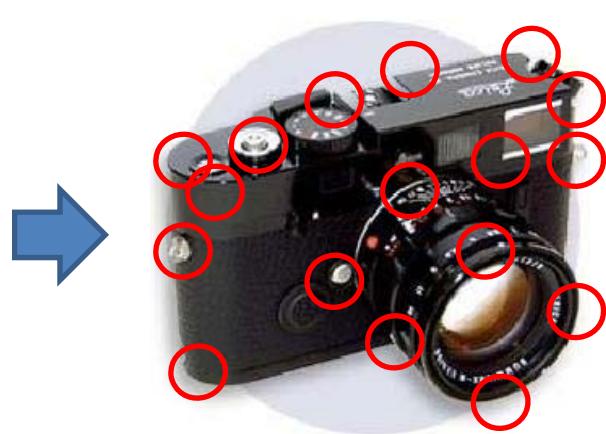
カテゴリ認識の流れ



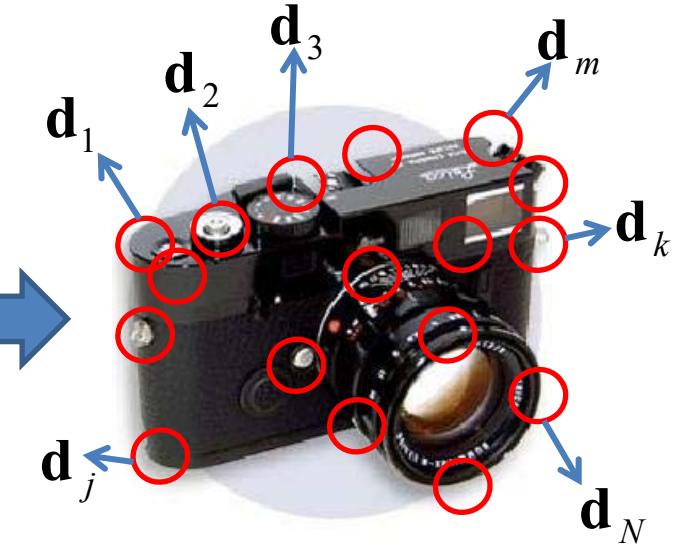
カテゴリ認識のパイプライン



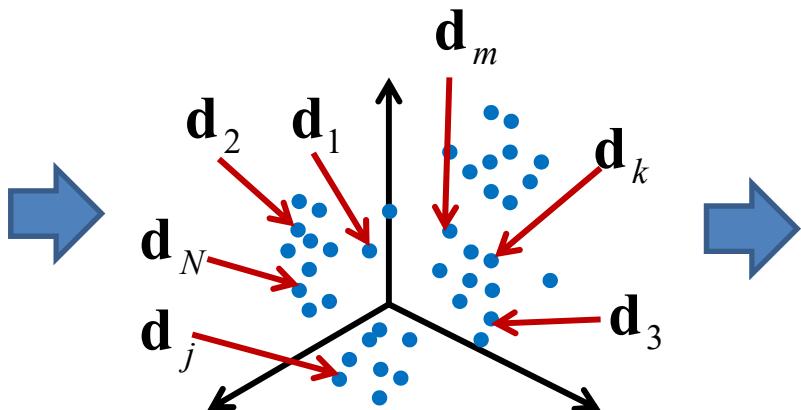
1) Input Image



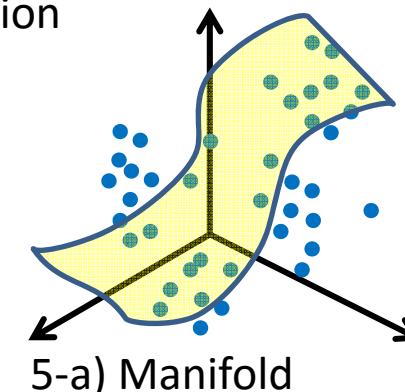
2) Detection



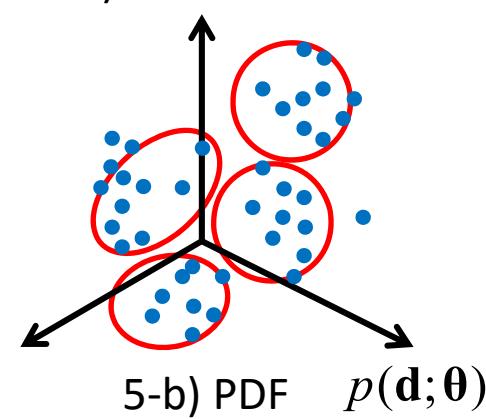
3) Description



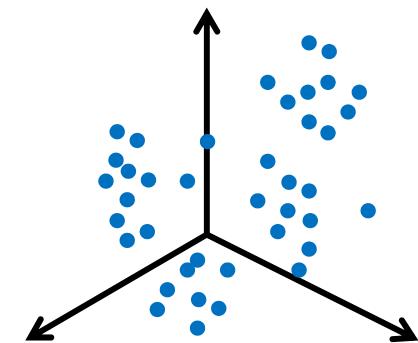
4) Local descriptors in feature space



5-a) Manifold



5-b) PDF $p(\mathbf{d}; \theta)$



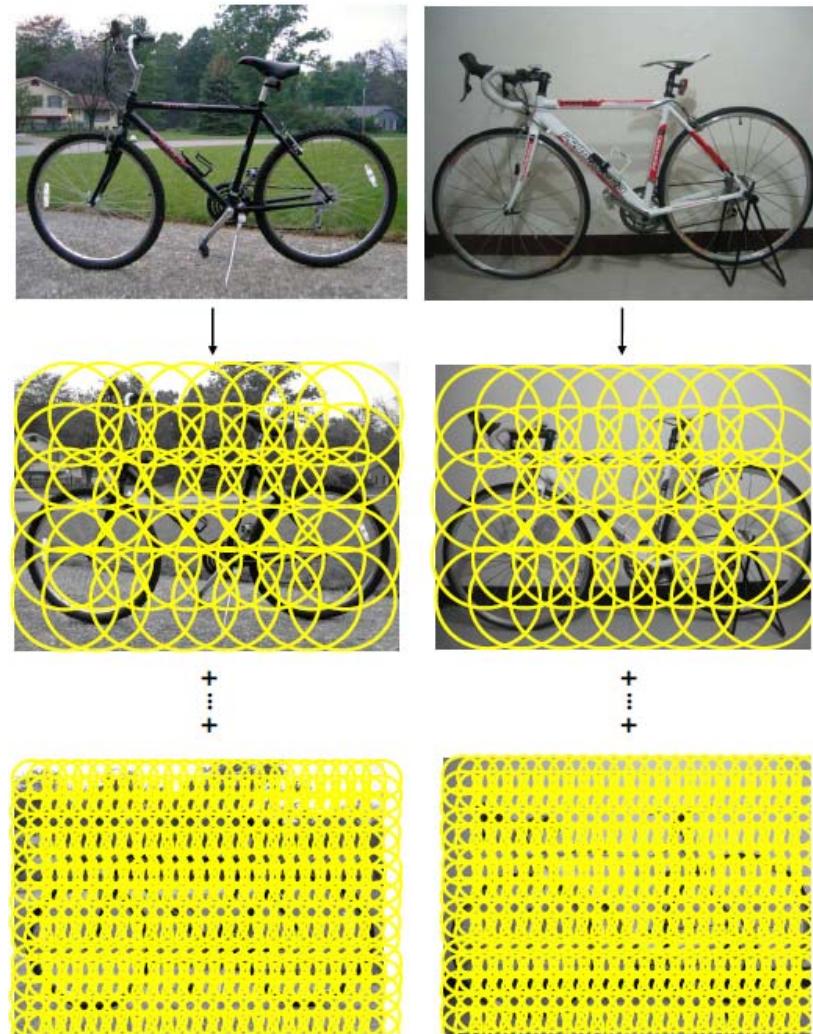
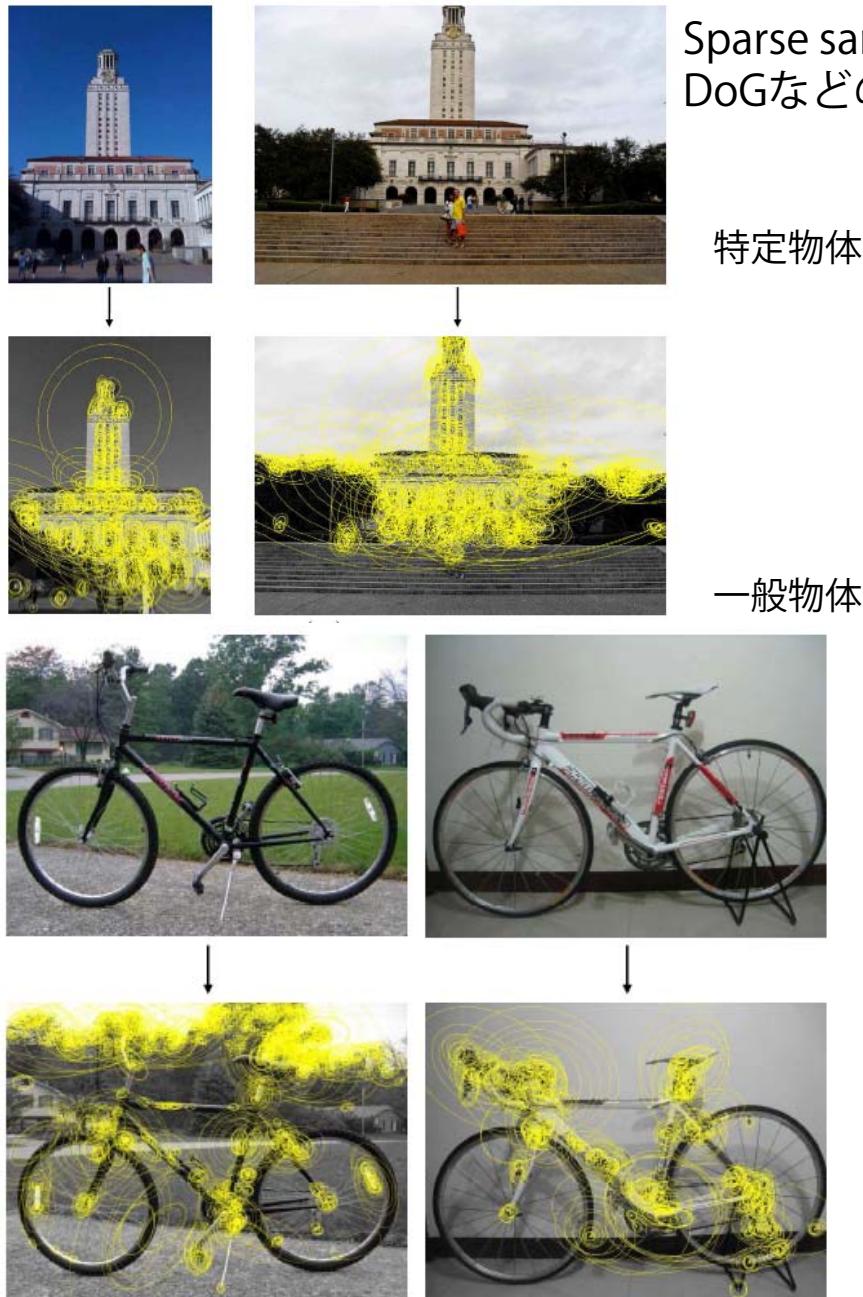
5-c) raw

局所特徴の直接的利用

Detection: sampling

K. Grauman and B. Leibe

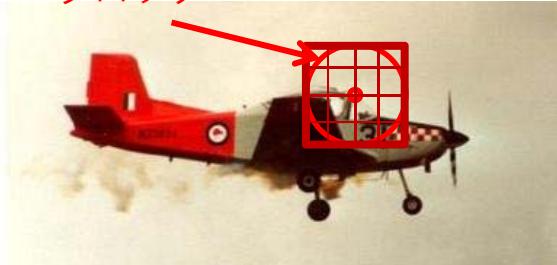
Dense sampling
グリッド上など



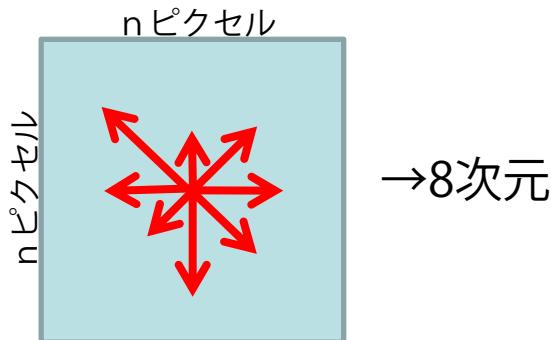
特定物体には検出器は（一般的に）有効
一般物体には密なサンプリングが（一般的に）有効 6

SIFT: 局所記述子

ブロック



1. ブロックを表現するベクトル



3. 16ブロックの勾配ヒストグラムをまとめて一つのベクトルとする

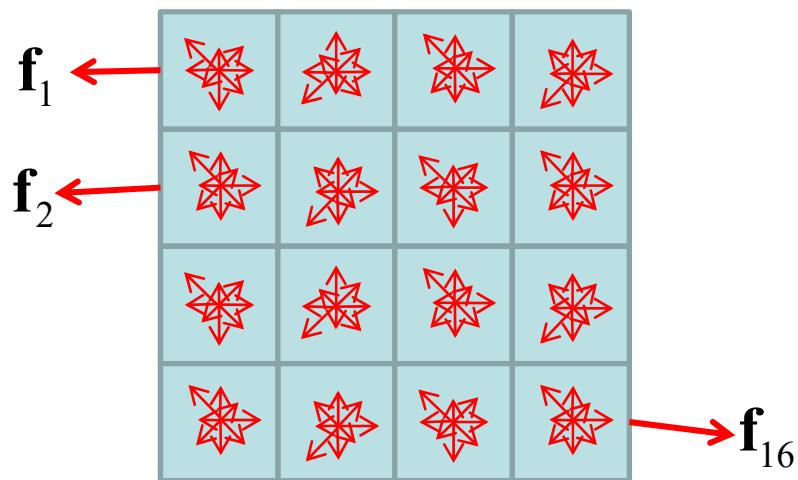
$$\mathbf{f}^T = \left(\mathbf{f}_1^T \ \mathbf{f}_2^T \ \cdots \ \mathbf{f}_{16}^T \right)$$

つまり、8次元×16ブロック=128次元のベクトルとなる

(注) ここまでプロセスをSIFT descriptorと呼ぶ。

SIFTとは特徴点検出+SIFT descriptorのことで、両者は最近では区別される

2. 着目領域内の16ブロック全てに勾配ヒストグラムを計算する



4. 得られたベクトルを正規化する

$$\mathbf{f}' = \frac{\mathbf{f}}{\|\mathbf{f}\|}$$

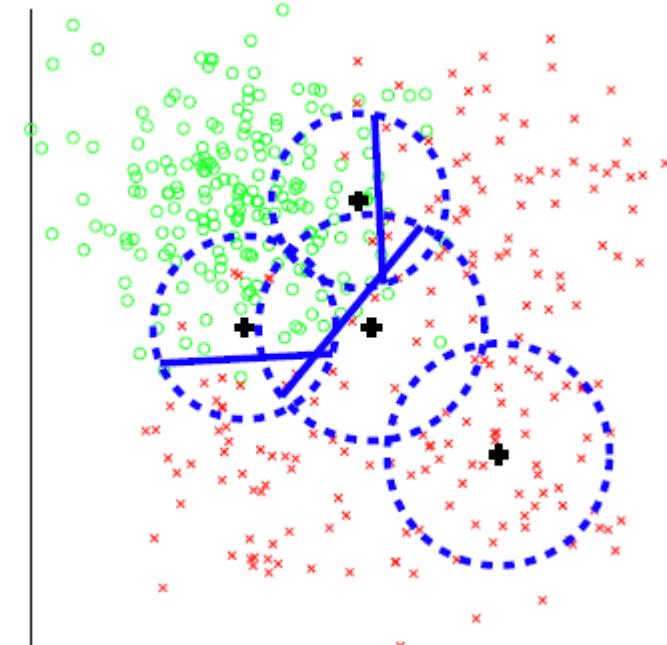
局所領域での正規化を行っているので
照明変化に頑健になる

Descriptor matching: SVM-KNN

H. Zhang, A. C. Berg, M. Maire, and J. Malik.

SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition.
In CVPR, 2006.

- Naïve version
 - クエリサンプルと全ての距離を計量し, k-NNを抽出
 - K-NNが全て同じクラスなら, そのクラスをクエリに付与して終了.
 - 全て同じクラスでなければ, k-NNでカーネル行列を作成し, kernel SVMの識別器を構成する.
 - 構成したkernel SVMでクエリを識別する.



画像間距離

Geometric Blur

$$D^A(I_L \rightarrow I_R) = \frac{1}{m} \sum_{i=1}^m \min_{j=1..n} \|F_i^L - F_j^R\|^2$$

$$D^A(I_L, I_R) = D^A(I_L \rightarrow I_R) + D^A(I_R \rightarrow I_L)$$

$$+ \lambda \sum_{k=1}^{n\text{filt}} \|h_k^L - h_k^R\|_{L1}$$

Texture Histogram

Naïve Bayes Nearest Neighbor

- O. Boiman, E. Shechtman, and M. Irani. In Defense of Nearest-Neighbor Based Image Classification. In CVPR, 2008.

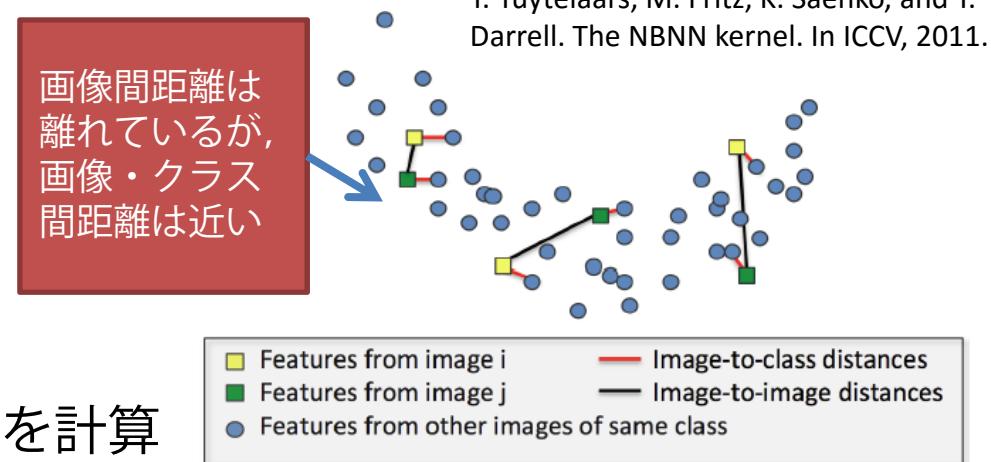
- 非常に単純だが高性能
- 画像-クラス間距離を利用

- アルゴリズム
 - クエリ画像から局所記述子を計算
 - クエリ画像の各局所記述子に関して、クラス内の全局所記述子の中で最近傍のものを探す

$$\text{NN}_C(d_i)$$

- クエリ画像の全局所記述子とクラス内の最近傍点とのユークリッド距離の総和を計算し、この距離が最も短いクラスにクエリ画像を割り当てる。

$$\hat{C} = \arg \min_C \sum_{i=1}^n \| d_i - \text{NN}_C(d_i) \|^2$$



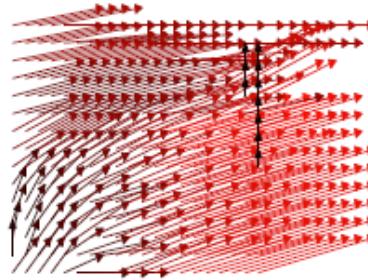
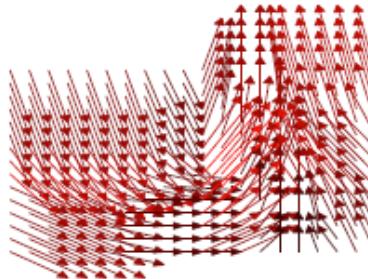
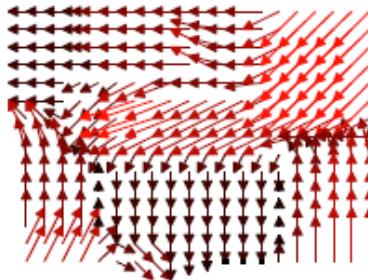
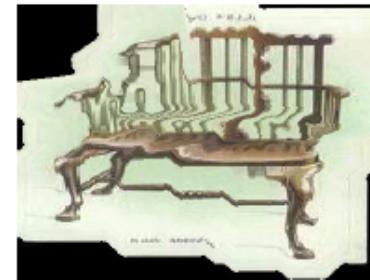
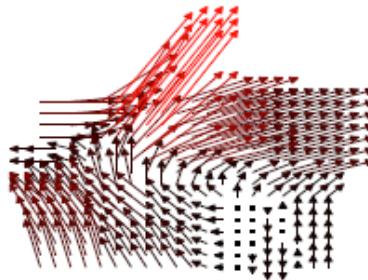
NBNN Kernel

- T. Tuytelaars, M. Fritz, K. Saenko, and T. Darrell. The NBNN kernel. In ICCV, 2011.

Algorithm 2: the NBNN kernel

1. Compute a set of features $X = \{\mathbf{x}\}$.
2. $\forall \mathbf{x} \forall c$ Compute the NN of \mathbf{x} in c : $NN^c(\mathbf{x})$,
and its distance-to-class $d_{\mathbf{x}}^c = \|\mathbf{x} - NN^c(\mathbf{x})\|^2$.
3. $\forall c$ $\Phi^c(X) = \sum_{\mathbf{x} \in X} f(d_{\mathbf{x}}^1, \dots, d_{\mathbf{x}}^{|C|})$.
4. $\Phi(X) = [\Phi^1(X) \dots \Phi^{|C|}(X)]^T$.
5. Repeat steps 1-4 for a second set of features $Y = \{\mathbf{y}\}$.
6. $K(X, Y) = \Phi(X)^T \Phi(Y)$.

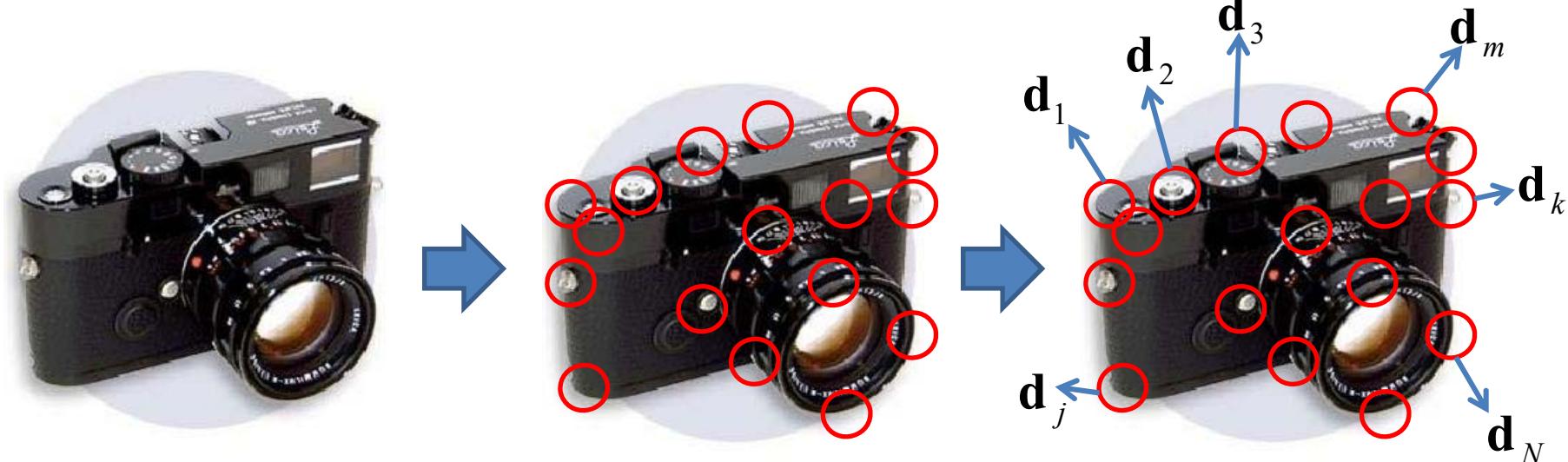
Graph Matching Kernel



O. Duchenne , A. Joulin and J. Ponce. A Graph-Matching Kernel for Object Categorization. ICCV, 2011.

確率密度分布によるコーディング

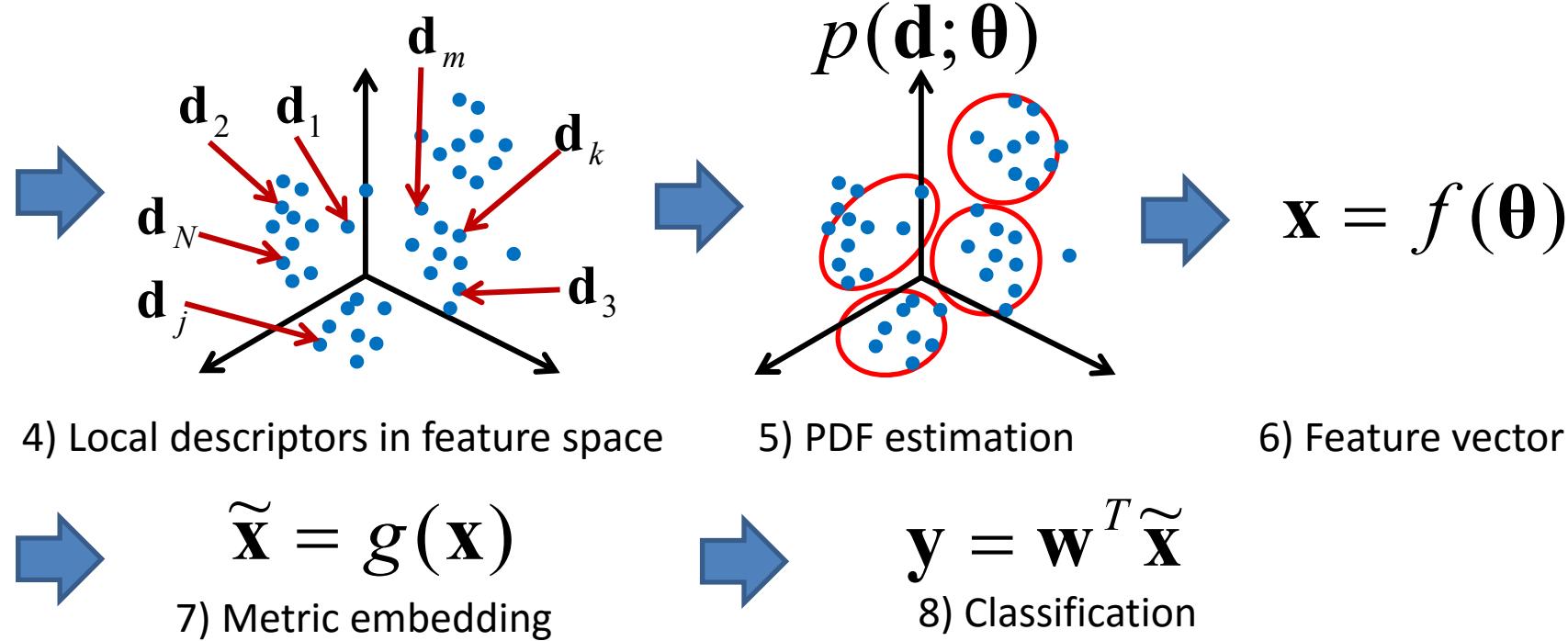
PDFを利用したカテゴリ識別のパイプライン



1) Input Image

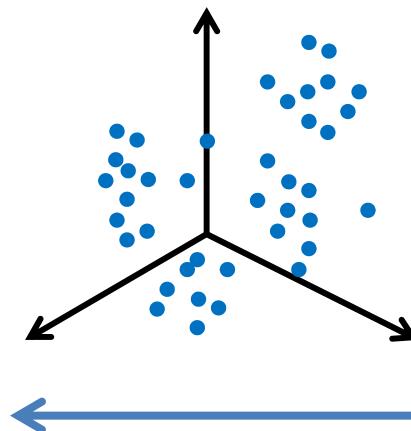
2) Detection

3) Description



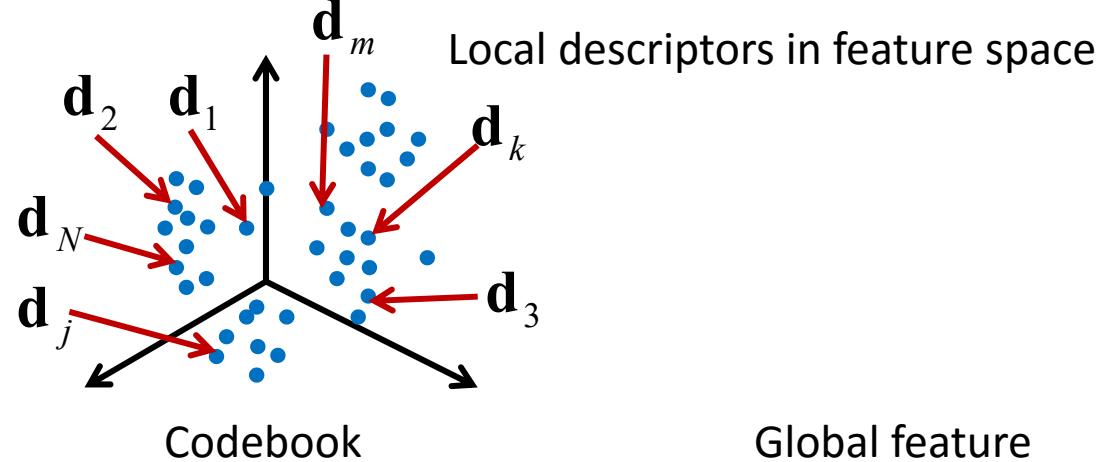
画像表現

Descriptor matching



of anchor points: large
Computational complexity: large

SVM-KNN
Naïve Bayes Nearest Neighbor
Graph Matching Kernel



of anchor points: small
Computational complexity: small

Bag of Visual Words
Gaussian Mixture Model
ScSPM, Super Vector, LLC
Fisher Vector

HLAC
GLC
Global Gaussian

大域特徴量による認識 Bag of Visual Words

- 画像認識のデファクトスタンダード的な画像特徴
- 局所特徴から大域特徴を作る枠組み
- 文章特徴とのアナロジー
 - Bag of Words
 - 単語の並び順、文法などを考慮しない
 - 例) 文章中で出てきた単語のヒストグラム
- テキストによる認識、情報検索技術が画像認識や検索にそのまま適応可能となる
 - ある意味、現在の画像認識技術の多くは、文章認識、検索技術からの借り物でできている。

Bag of Words?

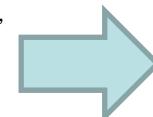
- The two hit it off quickly — unusual for the 6-year-old, who has autism — and the boy is imitating his playmate's every move, now nodding his head, now raising his arms.
- "Like Simon Says," says the autistic boy's mother, seated next to him on the floor.
- Yet soon he begins to withdraw; in a video of the session, he covers his ears and slumps against the wall.
- But the companion, a three-foot-tall robot being tested at the University of Southern California, maintains eye contact and performs another move, raising one arm up high.
- Up goes the boy's arm — and now he is smiling at the machine.
- In a handful of laboratories around the world, computer scientists are developing robots like this one: highly programmed machines that can engage people and teach them simple skills, including household tasks, vocabulary or, as in the case of the boy, playing, elementary imitation and taking turns.
- So far, the teaching has been very basic, delivered mostly in experimental settings, and the robots are still works in progress, a hackers' gallery of moving parts that, like mechanical savants, each do some things well at the expense of others.
- Yet the most advanced models are fully autonomous, guided by artificial intelligence software like motion tracking and speech recognition, which can make them just engaging enough to rival humans at some teaching tasks.
- Researchers say the pace of innovation is such that these machines should begin to learn as they teach, becoming the sort of infinitely patient, highly informed instructors that would be effective in subjects like foreign language or in repetitive therapies used to treat developmental problems like autism.
- Several countries have been testing teaching machines in classrooms. South Korea, known for its enthusiasm for technology, is "hiring" hundreds of robots as teacher aides and classroom playmates and is experimenting with robots that would teach English.
- Already, these advances have stirred dystopian visions, along with the sort of ethical debate usually confined to science fiction. "I worry that if kids grow up being taught by robots and viewing technology as the instructor," said Mitchel Resnick, head of the Lifelong Kindergarten group at the Media Laboratory at the Massachusetts Institute of Technology, "they will see it as the master."
- Most computer scientists reply that they have neither the intention, nor the ability, to replace human teachers. The great hope for robots, said Patricia Kuhl, co-director of the Institute for Learning and Brain Sciences at the University of Washington, "is that with the right kind of technology at a critical period in a child's development, they could supplement learning in the classroom."

Robot: 7

Boy: 4

Year: 1

Computer: 2



...

...



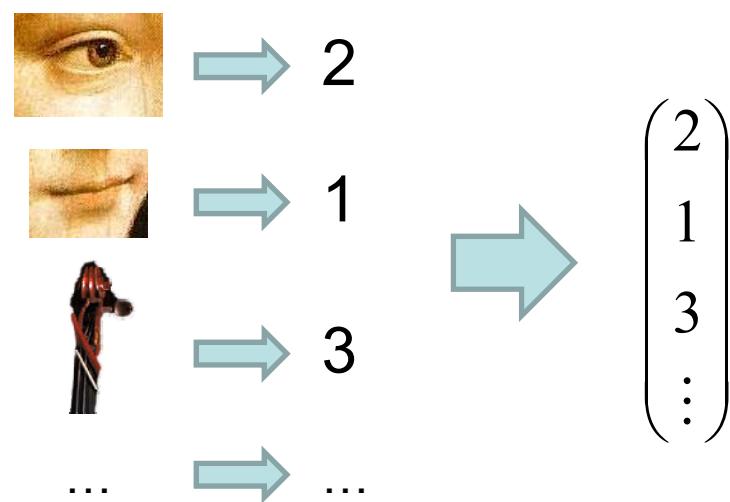
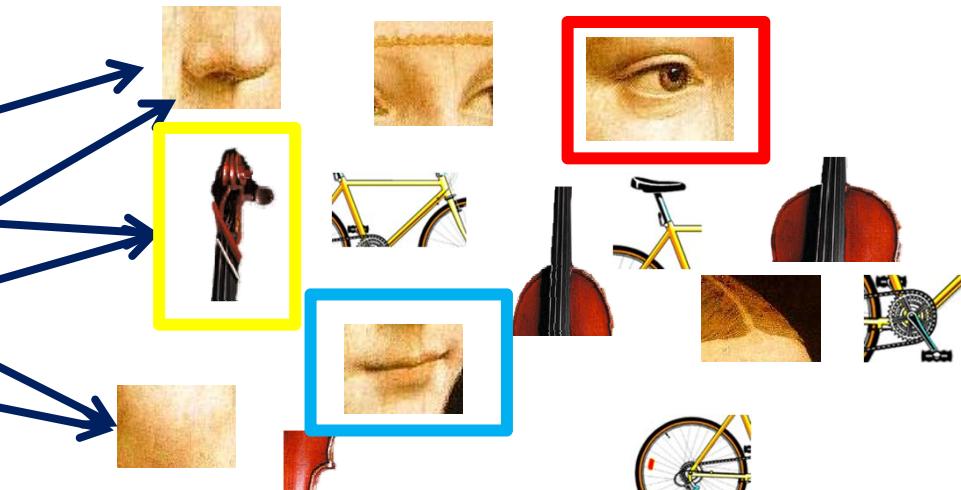
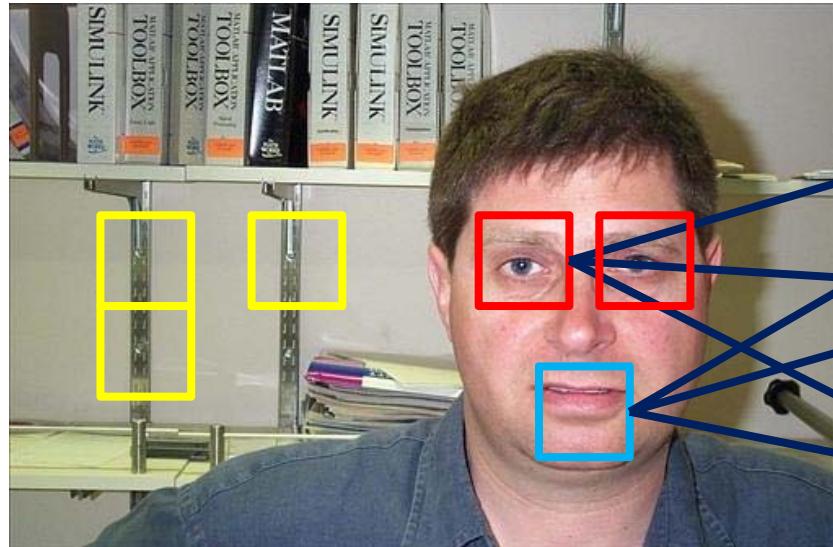
(7
4
1
2
⋮)

特徴ベクトル

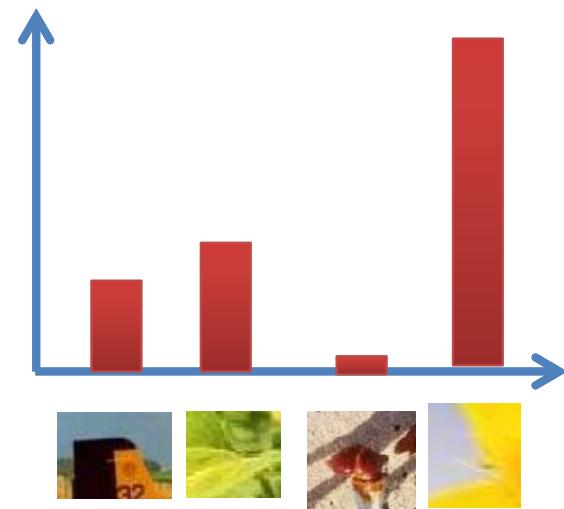
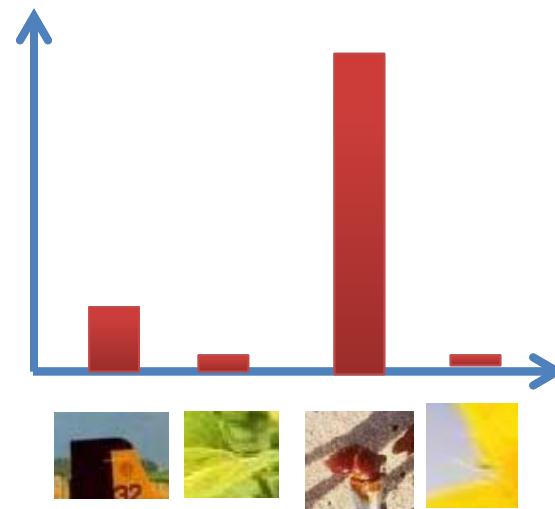
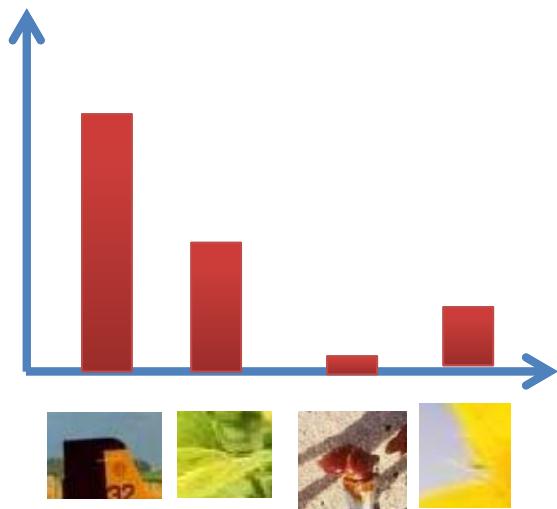
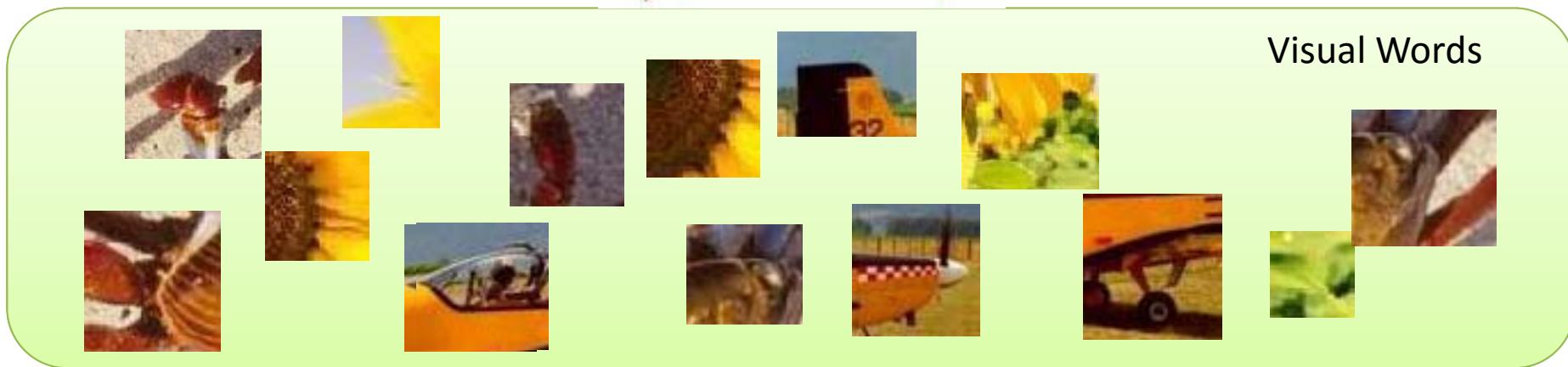
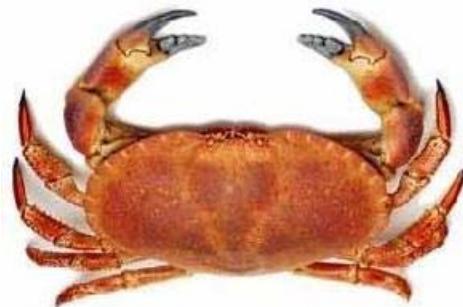
Visual Words?

Li Fei Fei, cvpr07 tutorial
より抜粋

Visual words



Bag of Visual Words?



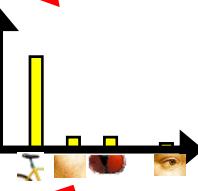
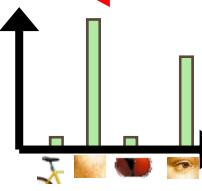
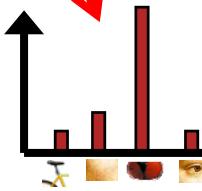
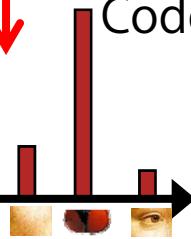
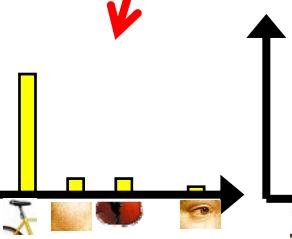
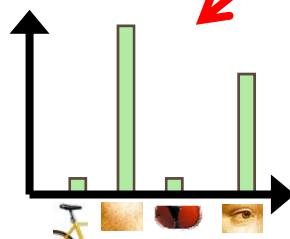
Bag of Visual Wordsによる認識

1. 訓練データからCode wordの作成



Code book

2. 訓練データのBoWを計算

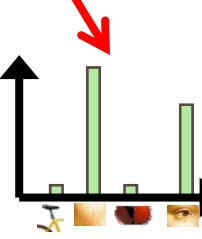
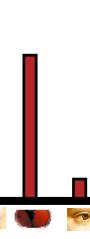


3. クラス識別器を構築

Classifier

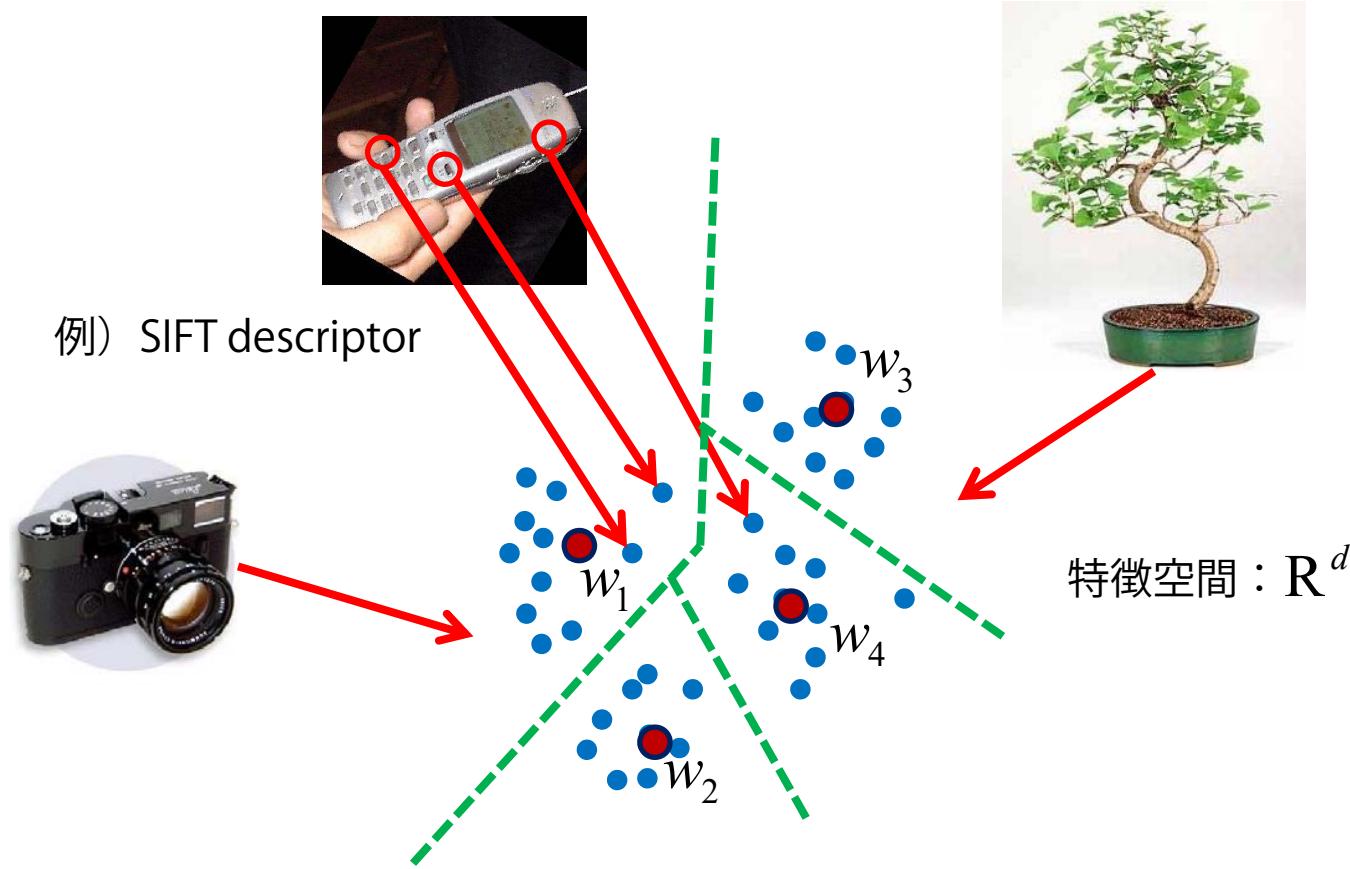
5. クラス識別器で識別

4. テストデータのBoWを計算



Bonsai, camera, cellphone

Code wordsの生成：clustering



- ベクトル量子化と呼ばれるプロセス
- 一般的にk-meansによるクラスタリング
 - 階層的クラスタリング: Vocabulary Tree
- 局所記述子にはSIFTがよく用いられる
 - もちろんSURFやRGB, Self Similarityでもよい

K-means

1. 特徴ベクトル集合 \mathcal{X} から K 個の初期クラスタの代表点 $\mathcal{C} = \{\mathbf{c}_i \in \mathbb{R}^d\}_{i=1}^K$ を選出する. 例えばランダムに K 個選択するなど.
2. \mathcal{X} から一つ特徴ベクトル \mathbf{x}_i を選択し, 全てのクラスタの代表点との距離を計算する. 最も距離の短いクラスタのラベルを特徴ベクトルに付与する.

$$k = \arg \min_j d(\mathbf{x}_i, \mathbf{c}_j) \Rightarrow \mathbf{x}_i \in C_k \quad (2)$$

3. 各クラスタの重心を求めてクラスタの代表点を更新する.

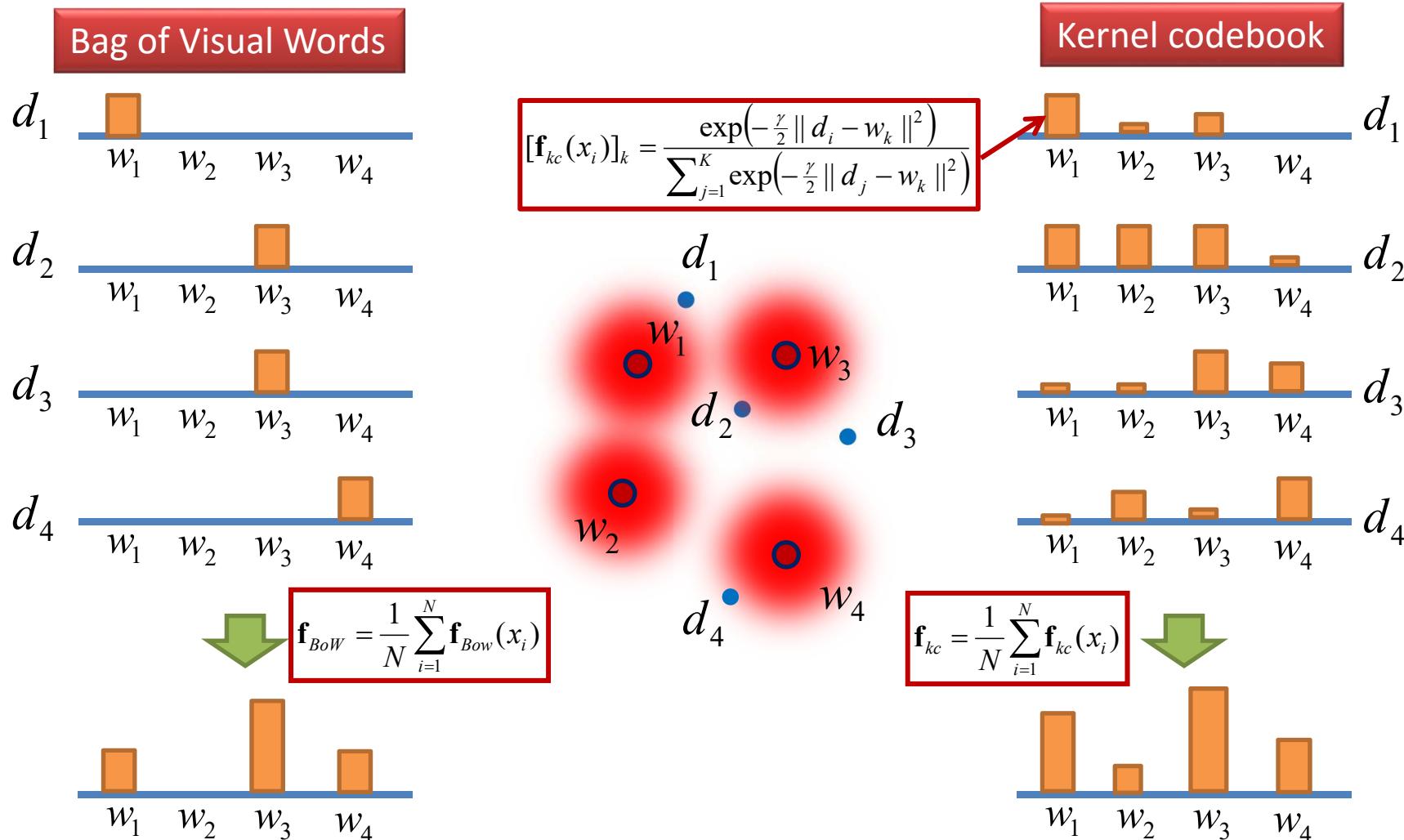
$$\mathbf{c}_k = \frac{1}{N_k} \sum_{j \in C_k} \mathbf{x}_j \quad (3)$$

ここで N_k はクラスタ k に属する特徴ベクトルの数である.

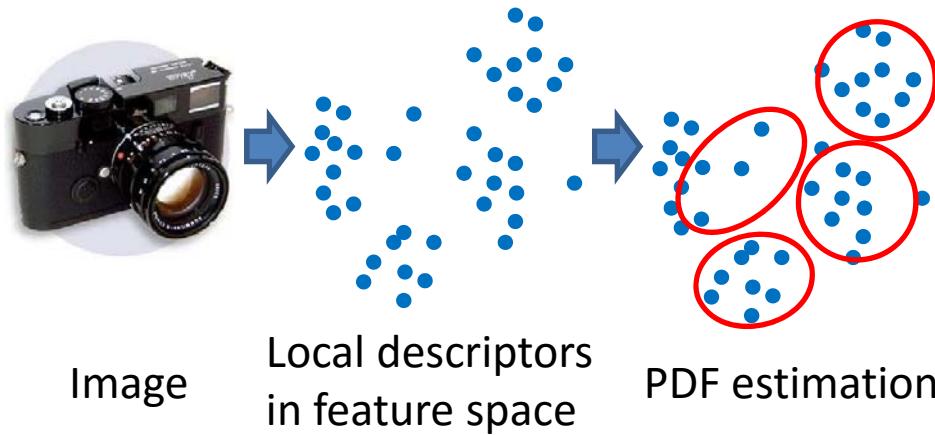
ステップ 2 と 3 を収束するまで繰り返す. ただし, このアルゴリズムはクラスタ代表点の初期値に依存して結果が変わる点に注意が必要である.

Kernel codebook

- 局所記述子を一つのコードワードに割り付けるのではなく、距離に応じた重み付けで全てのコードワードと関連づける。
- Jan C. van Gemert, Jan-Mark Geusebroek, Cor J. Veenman, and Arnold W.M. Smeulders. Kernel Codebooks for Scene Categorization. ECCV, 2008.



BoFのGMM利用による改善



$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \Sigma_k) = \sum_{k=1}^K \pi_k p_k(\mathbf{x})$$

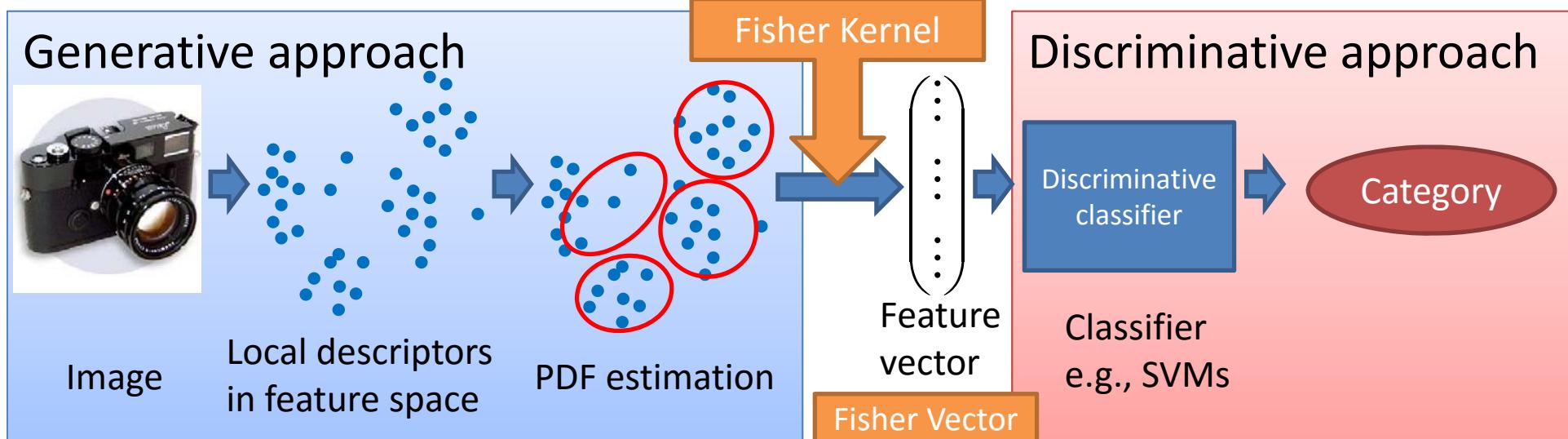
$$\gamma_n(k) = p(k | \mathbf{x}_n, \boldsymbol{\theta}^{(t)}) = \frac{\pi_k p_k(\mathbf{x}_n)}{\sum_{j=1}^K \pi_j p_j(\mathbf{x}_n)}$$

$$\mathbf{f} = \frac{1}{N} \sum_{n=1}^N [\gamma_n(1), \dots, \gamma_n(K)]^\top \in R^K$$

- メリット
 - 混合ガウス分布を構成する各ガウス分布がそれぞれ共分散を持つため、共分散を考慮した距離計量を利用できる
 - 混合ガウス分布では局所特徴と多くのコードワードとの関係を表現できるので、特徴空間における局所特徴の位置に関する情報をエンコードできる
- デメリット
 - 混合ガウス分布表現はBoF と比較してパラメータが多い
 - 混合ガウス分布 : $O(K(D^2/2 + D))$, BoF : $O(KD)$
 - 混合ガウス分布は訓練データに対して過剰適合する可能性があり、学習時に正則化を行う必要

フィッシャーベクトル

F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. CVPR, 2007.



- 混合ガウス分布を用いた確率密度分布推定によるBoF の改良
 - 生成モデル (generative model)
- 生成モデルを識別的なアプローチに適応可能なより洗練された手法があれば識別性能の改善につながる。
- フィッシャーカーネル (Fisher Kernel)
 - 生成的アプローチ (generative approach) と識別的アプローチ (discriminative approach) を結合させる強力な枠組み→確率分布の空間に適切な距離計量を埋め込む
 - 確率分布のなす空間は、Fisher 情報行列を計量とするリーマン空間
 - 手順
 1. 局所特徴を生成する確率密度分布から導出される勾配ベクトルの計算
 2. 画像を表現する一つの特徴ベクトルの計算
→ **フィッシャーベクトル (Fisher Vector)**
 3. 得られた特徴ベクトルを識別的分類機に入力する。

フィッシャーベクトルのメリット

- ・ 豊かな特徴ベクトル表現
 - BoF と比較してフィッシャーカーネルを利用する
メリットは、コードブックサイズが同じであれば
より要素数の多い特徴ベクトルが得られる。
 - コードブックサイズ : K, 局所特徴の次元 : d
 - BoFの次元 : K
 - フィッシャーベクトル : $(2d+1)K-1$
 - 特徴ベクトルの表現する情報が多いため計算コストの高いカーネル法を利用して高次元空間へ射影
する必要がなく、線形識別器でも十分な識別性能
を出すことが可能となる。



大規模データに
最も重要な要素

フィッシャーべクトル詳細

- 局所特徴群

$$\mathcal{X} = \{\mathbf{x}_n \in R^D\}_{n=1}^N$$

- あらゆる画像内容を表現する局所特徴の確率密度分布

$$u_\theta$$

Tommi Jaakkola and David Haussler. Exploiting Generative Models in Discriminative Classifiers. NIPS, 1998.

- 対数尤度の勾配

$$G_\theta^\mathcal{X} = \frac{1}{N} \nabla_\theta \log u_\theta(\mathcal{X}|\boldsymbol{\theta})$$

- データに最も適合するように確率密度関数のパラメータが修正すべき方向を表現
- 異なるデータサイズ集合をパラメータ数に依存した特定の長さの特徴ベクトルに変換
- 内積を利用する識別機には適切な計量が必要！！

- フィッシャー情報行列

$$F_\theta = E_X [\nabla_\theta \log u_\theta(\mathcal{X}|\boldsymbol{\theta}) \nabla_\theta \log u_\theta(\mathcal{X}|\boldsymbol{\theta})^\top]$$

- フィッシャーべクトル (Fisher Vector)

$$\mathcal{G}_\theta^\mathcal{X} = F_\theta^{-1/2} \nabla_\theta \log u_\theta(\mathcal{X}|\boldsymbol{\theta})$$

フィッシャー情報行列による対数尤度の勾配の正規化

混合ガウス分布におけるフィッシャーベクトル

- 確率密度分布を混合ガウス分布とする
 - 共分散行列は対角行列と仮定

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \Sigma_k) = \sum_{k=1}^K \pi_k p_k(\mathbf{x})$$

- 対数尤度の微分

$$G_\theta^\mathcal{X} = \frac{1}{N} \nabla_\theta \log u_\theta(\mathcal{X} | \theta)$$

画像1枚から得られる局所特徴の集合

あらゆる画像を生成する確率密度分布

負担率：局所特徴 x_n がGMMのコンポーネント k に属する確率

$$\frac{\partial \mathcal{L}(\mathcal{X} | \theta)}{\partial \pi_k} = \sum_{n=1}^N \left[\frac{\gamma_n(k)}{\pi_k} - \frac{\gamma_n(1)}{\pi_1} \right]$$

GMMのBoFとほぼ同じ

$$\mathbf{f} = \frac{1}{N} \sum_{n=1}^N [\gamma_n(1), \dots, \gamma_n(K)]^\top \in R^K$$

$$\frac{\partial \mathcal{L}(\mathcal{X} | \theta)}{\partial \boldsymbol{\mu}_k^d} = \sum_{n=1}^N \gamma_n(k) \left[\frac{\mathbf{x}_n^d - \boldsymbol{\mu}_k^d}{(\sigma_k^d)^2} \right]$$

局所特徴 x_n とGMMの各コンポーネント k の平均との差分

$$\frac{\partial \mathcal{L}(\mathcal{X} | \theta)}{\partial \sigma_k^d} = \sum_{n=1}^N \gamma_n(k) \left[\frac{(\mathbf{x}_n^d - \boldsymbol{\mu}_k^d)^2}{(\sigma_k^d)^3} - \frac{1}{\sigma_k^d} \right]$$

- 混合比：BoFとほぼ同じ
- 平均，分散：あらゆる画像を表現するpdfの平均との差分
- BoFは0次，Fisher Vectorは1次，2次の統計量を含む
- 分散の表現は平均の表現とあまり差がない？本来は各コンポーネント間の相関が必要

フィッシャー情報行列

- フィッシャー情報行列

$$F_\theta = E_X[\nabla_\theta \log u_\theta(\mathcal{X}|\boldsymbol{\theta}) \nabla_\theta \log u_\theta(\mathcal{X}|\boldsymbol{\theta})^\top]$$

- 混合ガウス分布において近似的に閉じた解が得られる
- 仮定
 - フィッシャー情報行列は対角行列
 - 共分散行列は対角行列
 - 負担率はピーキー
 - 一枚の画像から得られる局所特徴数は一定

フィッシャー情報行列の要素

$$\frac{\partial \mathcal{L}(\mathcal{X}|\theta)}{\partial \pi_k} \rightarrow f_{\pi_k} = N \left(\frac{1}{\pi_k} + \frac{1}{\pi_1} \right)$$
$$\frac{\partial \mathcal{L}(\mathcal{X}|\theta)}{\partial \mu_k^d} \rightarrow f_{\mu_k^d} = \frac{N\pi_k}{(\boldsymbol{\sigma}_k^d)^2}$$
$$\frac{\partial \mathcal{L}(\mathcal{X}|\theta)}{\partial \sigma_k^d} \rightarrow f_{\sigma_k^d} = \frac{2N\pi_k}{(\boldsymbol{\sigma}_k^d)^2}$$

フィッシャーベクトルの直感的解釈

http://www.image-net.org/challenges/LSVRC/2010/ILSVRC2010_XRCE.pdf

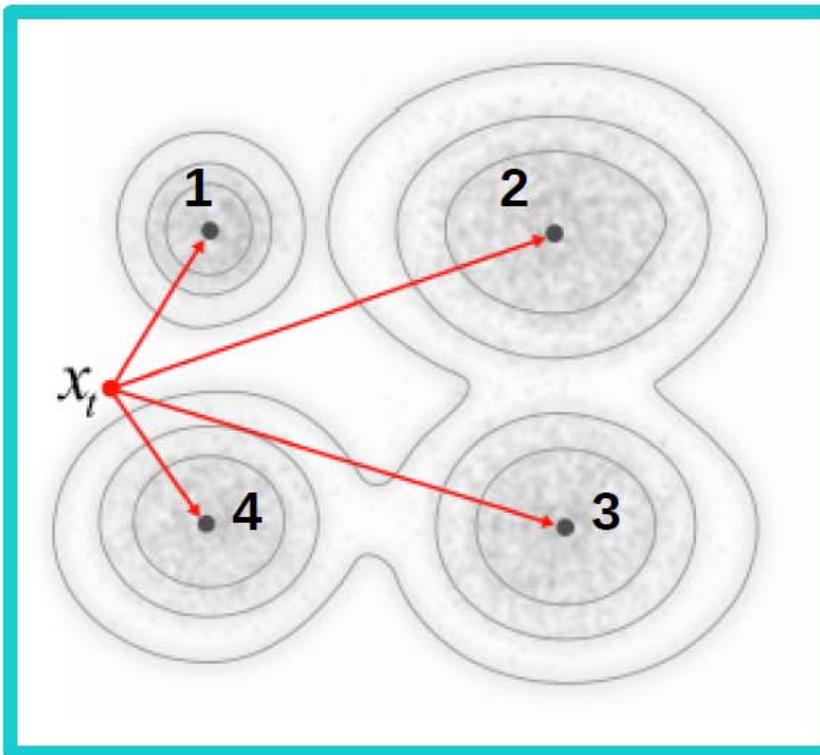
BOV

Hard Assignment

[0 0 0 1]

Soft Assignment

[.3 .1 .1 .5]



Fisher Vector

Gradient wrt w

[.15 -.2 -.35 .2]

Gradient wrt mean

[.8 -1.5 -3.7 -1.3 -3.8 1.2 -.9 1.4]

Gradient wrt var

[-1.2 -.9 1.4 -.8 1.5 -3.7 1.3 -3.8]

Bag of Visual Words (GMM)

$$\mathbf{f} = \frac{1}{N} \sum_{n=1}^N [\gamma_n(1), \dots, \gamma_n(K)]^\top \in R^K$$

Fisher Vector

$$\frac{\partial \mathcal{L}(\mathcal{X}|\theta)}{\partial \pi_k} = \sum_{n=1}^N \left[\frac{\gamma_n(k)}{\pi_k} - \frac{\gamma_n(1)}{\pi_1} \right]$$

$$\frac{\partial \mathcal{L}(\mathcal{X}|\theta)}{\partial \mu_k^d} = \sum_{n=1}^N \gamma_n(k) \left[\frac{\mathbf{x}_n^d - \mu_k^d}{(\sigma_k^d)^2} \right]$$

$$\frac{\partial \mathcal{L}(\mathcal{X}|\theta)}{\partial \sigma_k^d} = \sum_{n=1}^N \gamma_n(k) \left[\frac{(\mathbf{x}_n^d - \mu_k^d)^2}{(\sigma_k^d)^3} - \frac{1}{\sigma_k^d} \right] \quad 29$$

フィッシャーベクトルの改善

- フィッシャーベクトルはBoFと比較して豊かな表現
 - しかしながら、そのまま画像識別に利用しても BoF とさほど性能に差がない。

$$\frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \pi_k} = \sum_{n=1}^N \left[\frac{\gamma_n(k)}{\pi_k} - \frac{\gamma_n(1)}{\pi_1} \right]$$

GMMのBoFとほぼ同じ

$$\frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k^d} = \sum_{n=1}^N \gamma_n(k) \left[\frac{\mathbf{x}_n^d - \boldsymbol{\mu}_k^d}{(\sigma_k^d)^2} \right]$$

局所特徴 \mathbf{x}_n とGMMの各コンポーネント k の平均との差分

$$\frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \sigma_k^d} = \sum_{n=1}^N \gamma_n(k) \left[\frac{(\mathbf{x}_n^d - \boldsymbol{\mu}_k^d)^2}{(\sigma_k^d)^3} - \frac{1}{\sigma_k^d} \right]$$

- 改善方法
 - L2正規化
 - パワー正規化
 - 空間ピラミッドの導入
- F. Perronnin, J. Sanchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. ECCV, 2010.

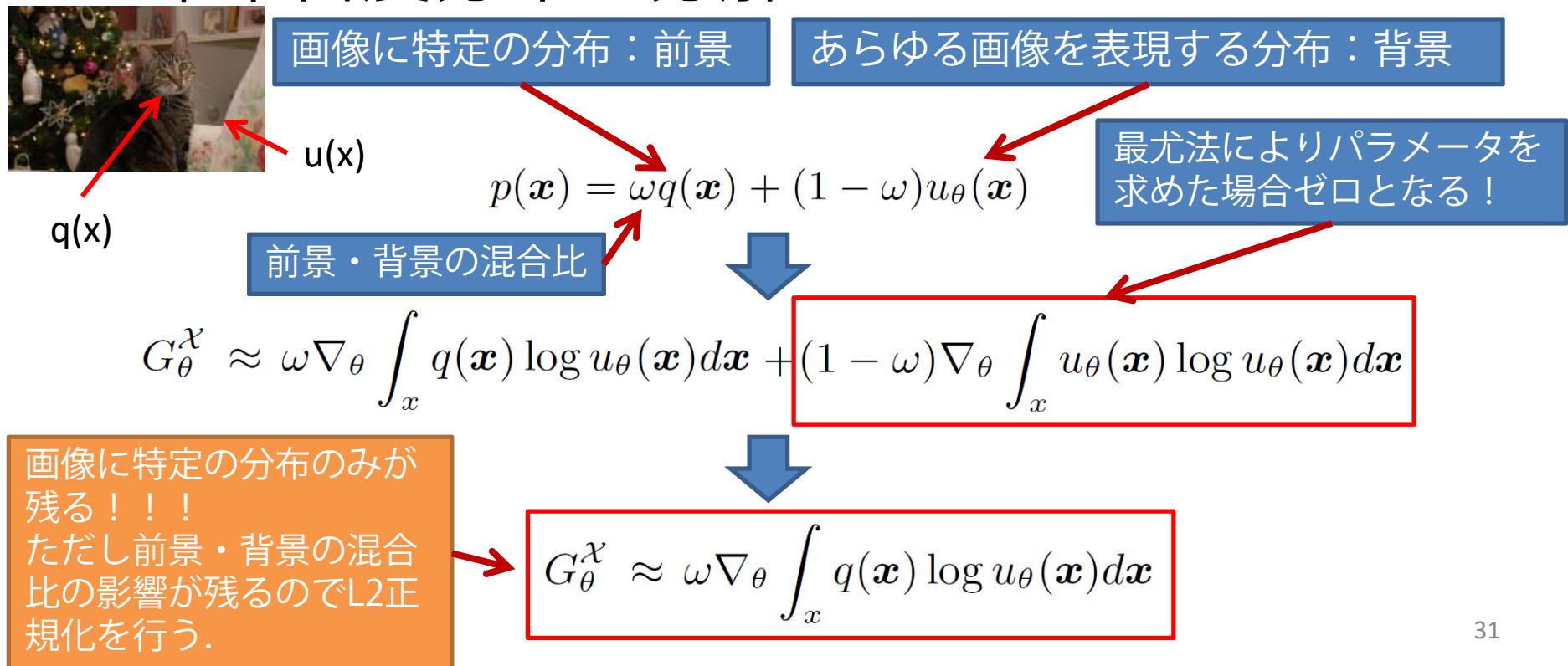
L2正規化によるフィッシャーベクトルの改善

- 対数尤度の勾配

$$G_{\theta}^{\mathcal{X}} = \frac{1}{N} \nabla_{\theta} \log u_{\theta}(\mathcal{X} | \theta) \longrightarrow G_{\theta}^{\mathcal{X}} \approx \nabla_{\theta} \int_x p(\mathbf{x}) \log u_{\theta}(\mathbf{x}) d\mathbf{x}$$

1枚の画像から得られた局所特徴群 \mathcal{X} は $p(\mathbf{x})$ に従うとする

- 確率密度分布の分解

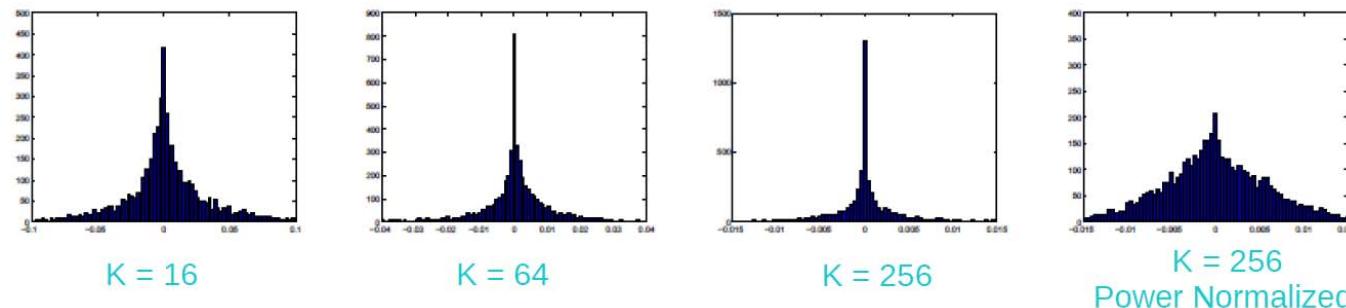


パワー正規化、空間ピラミッドによる フィッシャーベクトルの改善

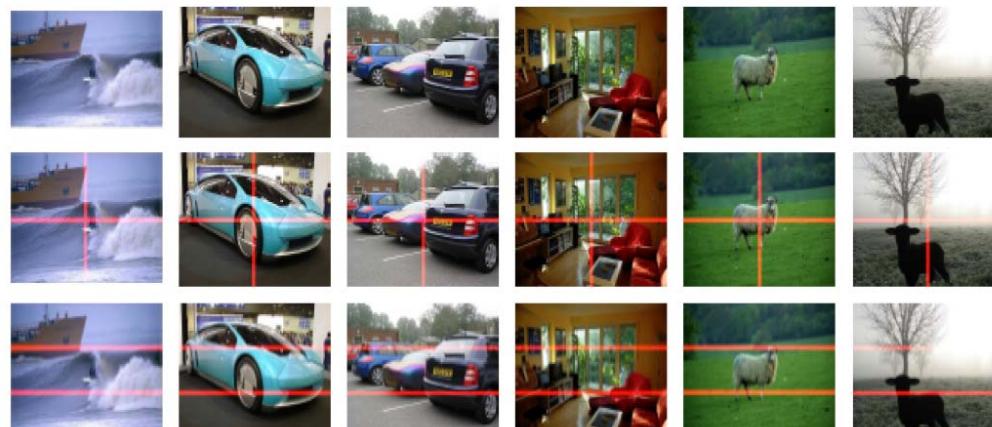
http://www.image-net.org/challenges/LSVRC/2010/ILSVRC2010_XRCE.pdf

- パワー正規化
 - 混合数の増加に伴いフィッシャーベクトルがスパースになる
 - スパースベクトルにおけるL2距離は性能が悪い
 - 方針1：カーネル法は計算コストが高い
 - 方針2：スパースにしない

$$f(z) = \text{sign}(z)|z|^\alpha$$



- 空間ピラミッド



画像1枚あたり8個
のフィッシャーベクトルを抽出

フィッシャーベクトルの性能

http://www.image-net.org/challenges/LSVRC/2010/ILSVRC2010_XRCE.pdf

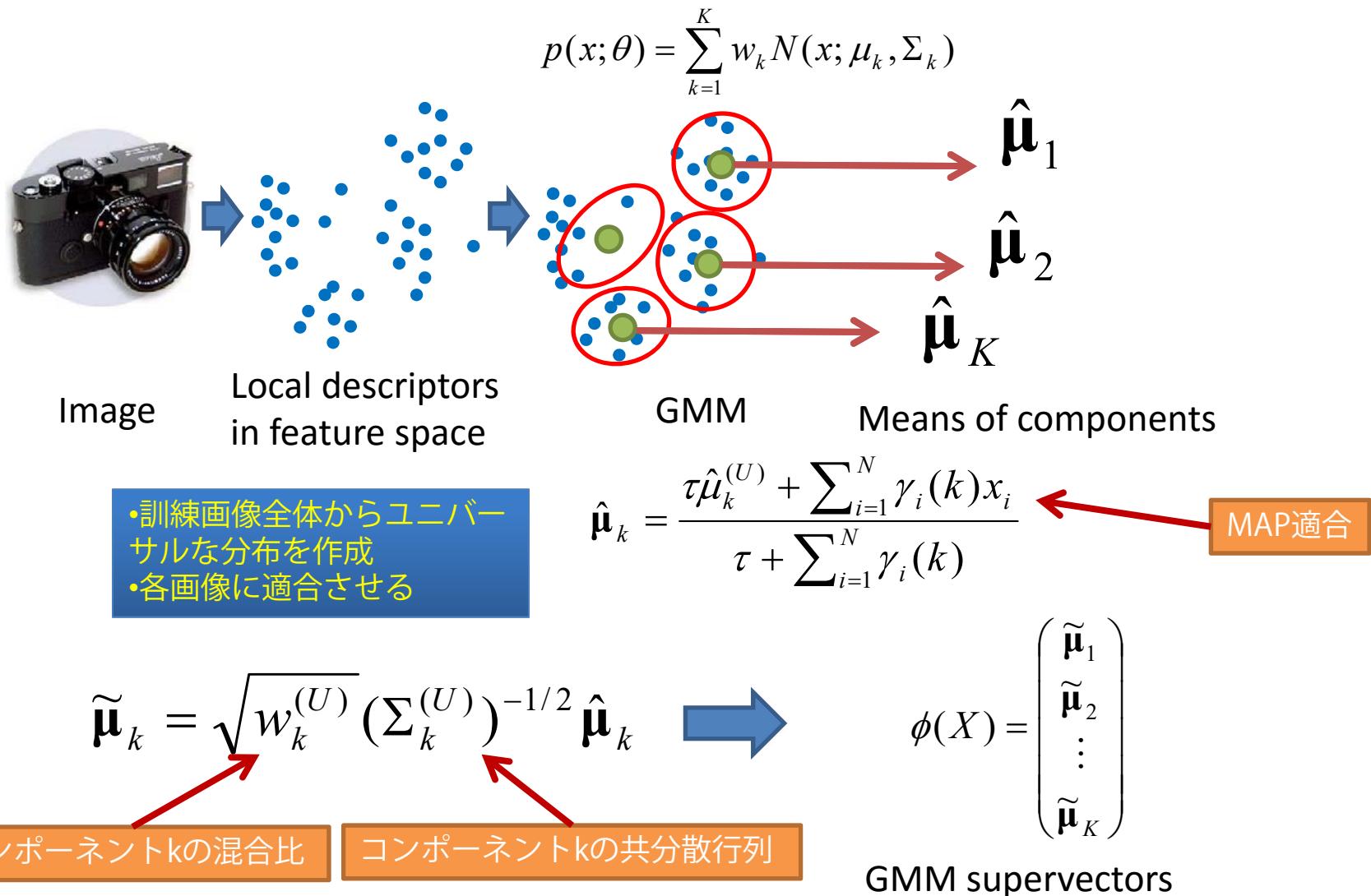
- Pascal VOC 2007
- 改良されたフィッシャーベクトルを利用
- 識別機：線形SVM

PN	L2	SP	SIFT	Col	S+C
-	-	-	47.9	34.2	45.9
✓	-	-	54.2	45.9	57.6
-	✓	-	51.8	40.6	53.9
-	-	✓	50.3	37.5	49.0
✓	✓	✓	58.3	50.9	60.3

パワー正規化>L2正規化>空間ピラミッド、の順で改善の効果が高い
33

GMM Supervectors

- W. M. Campbell and D. E. Sturim and D. A. Reynolds. Support vector machines using GMM supervectors for speaker verification. IEEE Signal Processing Letters, Vol.13, pp.308-311, 2006.
- もともと音声認識で利用されていたもの。



GMM Supervectors

- W. M. Campbell and D. E. Sturim and D. A. Reynolds. Support vector machines using GMM supervectors for speaker verification. IEEE Signal Processing Letters, Vol.13, pp.308-311, 2006.

GMM
supervectors

$$\hat{\mu}_k = \frac{\tau \hat{\mu}_k^{(U)} + \sum_{i=1}^N \gamma_i(k) x_i}{\tau + \sum_{i=1}^N \gamma_i(k)}$$

$$\begin{aligned}\tilde{\mu}_k &= \sqrt{w_k^{(U)}} (\Sigma_k^{(U)})^{-1/2} \hat{\mu}_k \\ &\approx \frac{\sqrt{w_k^{(U)}}}{\sum_{i=1}^N \gamma_k(i)} (\Sigma_k^{(U)})^{-1/2} \sum_{i=1}^N \gamma_i(k) x_i \\ &\approx \frac{1}{N \sqrt{w_k^{(U)}}} (\Sigma_k^{(U)})^{-1/2} \sum_{i=1}^N \gamma_i(k) x_i\end{aligned}$$

$Nw_k = \sum_{i=1}^N \gamma_i(k)$

GMM supervectorとFisher Vector
の平均成分はほぼ同一

Fisher Vectorの
平均成分

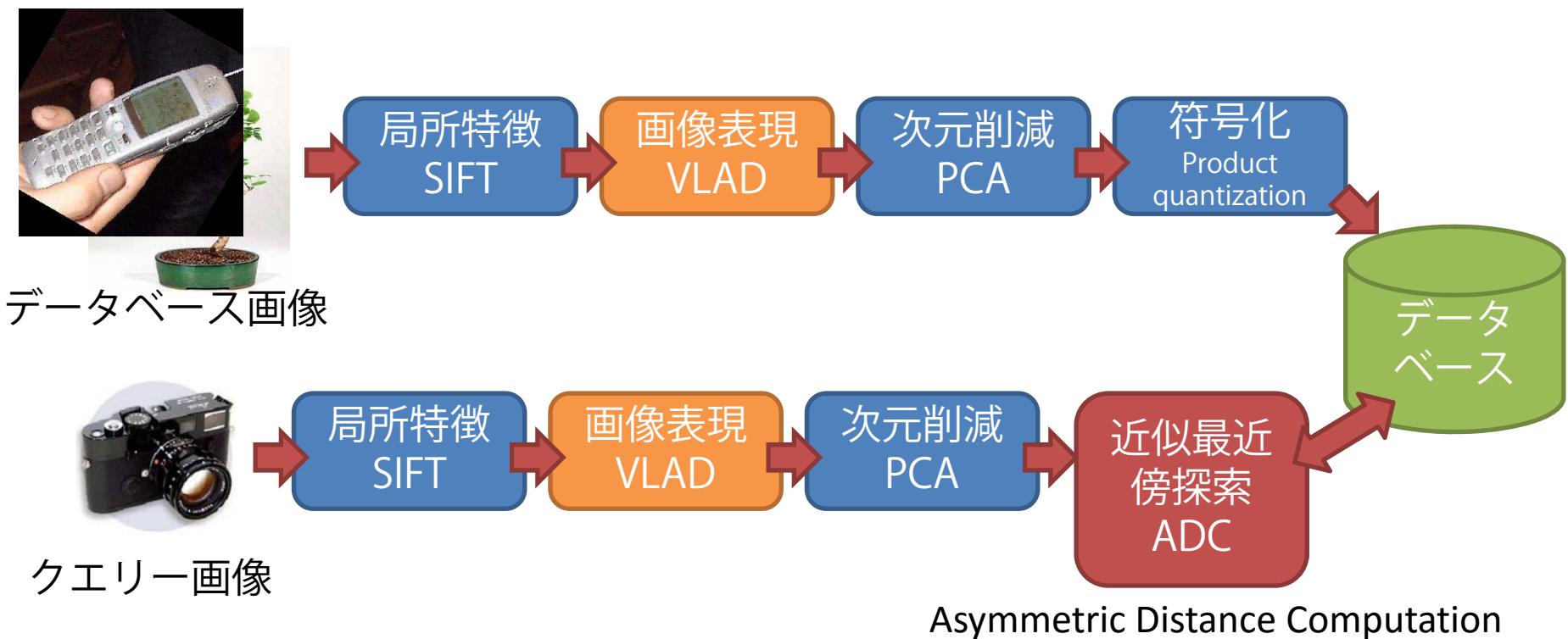
$$g_{\mu,i} = \frac{1}{N \sqrt{w_k}} \sum_{i=1}^N \gamma_i(k) (\Sigma_k)^{-1/2} (\mathbf{x}_i - \mu_k)$$

TRECVID 2011ではGMM supervectorの局所特徴の各
コンポーネントへの割り付けを高速化させることで
第一位の性能を上げている。

N. Inoue and K. Shinoda. A Fast MAP
Adaptation Technique for
GMMsupervector-based Video Semantic
Indexing. ACM Multimedia, 2011.

フィッシュシャーベクトルの 画像検索への応用例

- H. Jegou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. CVPR, 2010.
- 20bitに画像表現しても、生のBoFを使った検索と同じ検索性能
- パイプライン



VLAD

H. Jegou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. CVPR, 2010.

- Vector of Locally Aggregated Descriptors

$$z_i^d = \sum_{x \in \mathcal{X}_i} (\mathbf{x}^d - \mathbf{v}_i^d)$$

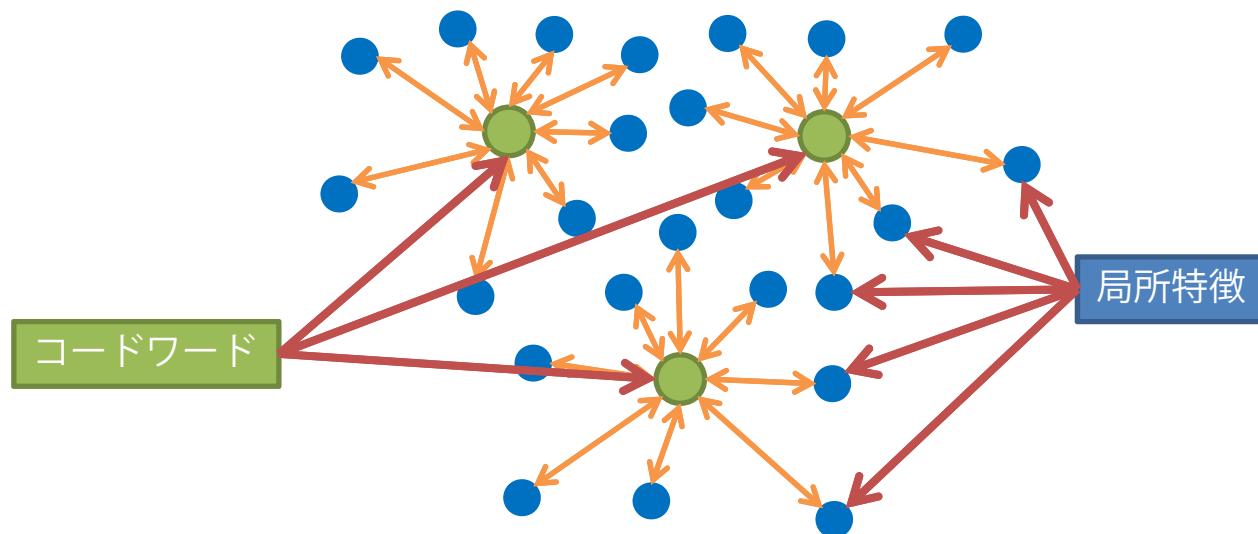
VLADのd番目要素

局所特徴のd番目要素

局所特徴が割り当てられたコードワードiのベクトル

この後 L2正規化

コードワードiに属する局所特徴集合



VLADとフィッシャーベクトル

・ フィッシャーベクトル

$$\frac{\partial \mathcal{L}(\mathcal{X}|\theta)}{\partial \pi_k} = \sum_{n=1}^N \left[\frac{\gamma_n(k)}{\pi_k} - \frac{\gamma_n(1)}{\pi_1} \right]$$

GMMのBoFとほぼ同じ

$$\frac{\partial \mathcal{L}(\mathcal{X}|\theta)}{\partial \mu_k^d} = \sum_{n=1}^N \gamma_n(k) \left[\frac{\mathbf{x}_n^d - \mu_k^d}{(\sigma_k^d)^2} \right]$$

局所特徴 \mathbf{x}_n とGMMの各コンポーネント k の平均との差分

$$\frac{\partial \mathcal{L}(\mathcal{X}|\theta)}{\partial \sigma_k^d} = \sum_{n=1}^N \gamma_n(k) \left[\frac{(\mathbf{x}_n^d - \mu_k^d)^2}{(\sigma_k^d)^3} - \frac{1}{\sigma_k^d} \right]$$

・ VLAD

$$z_i^d = \sum_{x \in \mathcal{X}_i} (\mathbf{x}^d - \mathbf{v}_i^d)$$

VLADのd番目要素

局所特徴のd番目要素

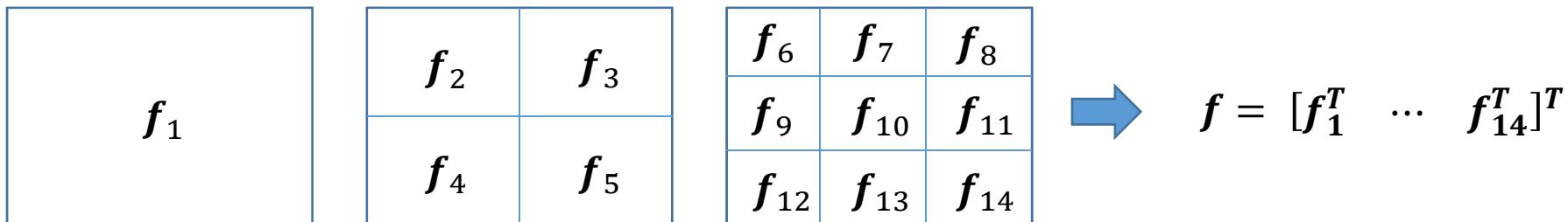
局所特徴が割り当てられたコードワード*i*のベクトル

コードワード*i*に属する局所特徴集合

負担率：ハードな割り当て
分散：全てのコンポーネントで同じ
→VLADはフィッシャーベクトルの平均に関する要素と同じ。
• (注) 分散を考えていないのでフィッシャーとは言い難い。

Improvement of VLAD

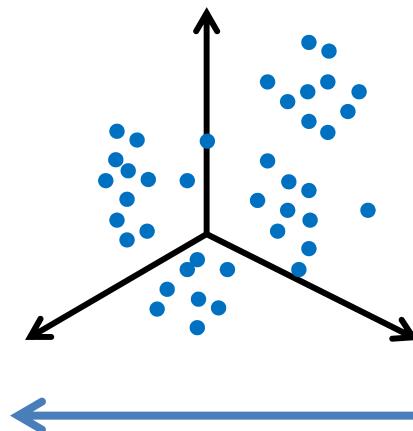
- All about VLAD. CVPR, 2013.
- Intra-normalization
 - Signed squared root (SSR) ($sign(x_i)\sqrt{|x_i|}$) + L2 normalizationではbursty visual featuresに過大な重みがかかる.
 - そこで、各VLADブロックに独立にL2 normalizationを行う.
- Multi-VLAD
 - 画像を領域に分割して、各領域からVLADを計算して、それらを結合して一つの特徴ベクトルとする.



- Vocabulary adaptation
 - データセットAで学習した辞書でデータセットBの特徴を表現すると、データセットBで学習した辞書と比較して性能が劣る.
 - (1)同じクラスタkに属する全ての局所記述子の平均を計算.
(2)再計算された平均を利用して、すべてのVLADを再計算する。この計算には局所記述子の総和のみを保持しておけばよい。

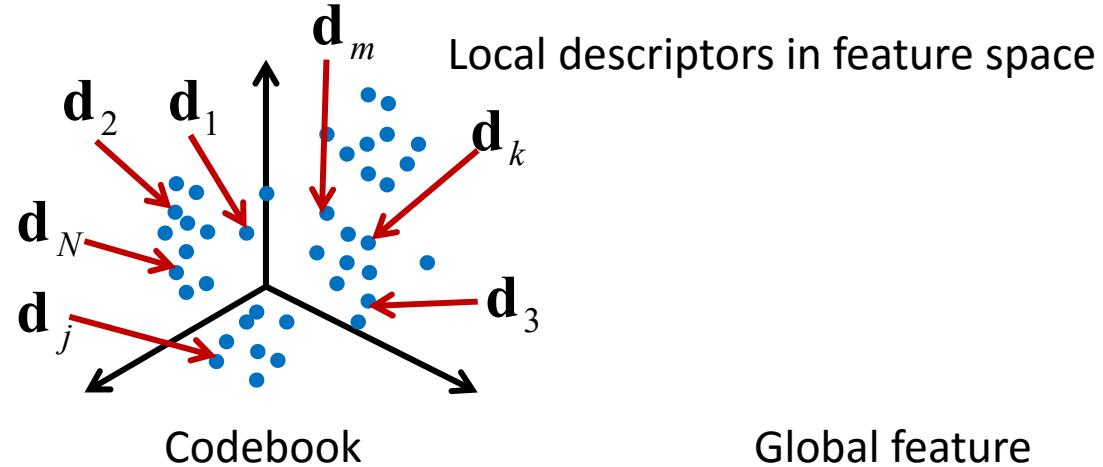
画像表現

Descriptor matching



of anchor points: large
Computational complexity: large

SVM-KNN
Naïve Bayes Nearest Neighbor
Graph Matching Kernel



of anchor points: small
Computational complexity: small

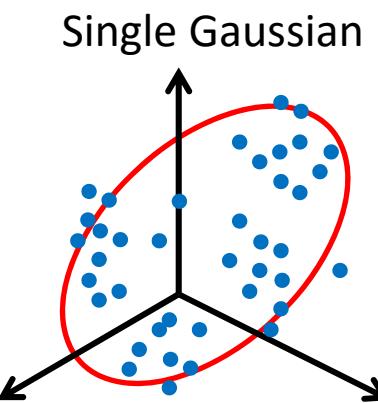
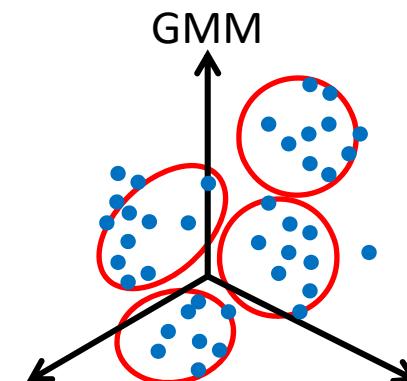
Bag of Visual Words
Gaussian Mixture Model
ScSPM, Super Vector, LLC
Fisher Vector

HLAC
GLC
Global Gaussian

Generalized Local Correlation (GLC)

H. Nakayama, T. Harada, and Y. Kuniyoshi. Dense Sampling Low-Level Statistics of Local Features. In CIVR, 2009.

- GMM
 - 表現能力が高い
 - GMMはパラメータが多いので、共分散行列の非対角成分を0とする場合が多い。
 - 計算コストが高い
- Single Gaussian
 - 表現能力に限界有り
 - パラメータが少ないので、共分散行列推定可能
 - 共分散行列の非対角成分を有効活用
 - 計算コストが低い



局所記述子

平均

$$\mu^{(j)} = \frac{1}{p^{(j)}} \sum_k^{p^{(j)}} \mathbf{v}_k^{(j)}$$

自己相関行列

$$R^{(j)} = \frac{1}{p^{(j)}} \sum_k^{p^{(j)}} \mathbf{v}_k^{(j)} \mathbf{v}_k^{(j)T}$$

GLC

$$\mathbf{x}^{(j)} = \begin{pmatrix} \mu^{(j)} \\ upper(R^{(j)}) \end{pmatrix}$$

Generalized Local Correlation (GLC)

H. Nakayama, T. Harada, and Y. Kuniyoshi. Dense Sampling Low-Level Statistics of Local Features. In CIVR, 2009.

- GLCは単純であるが結構いける

Table 2: Comparison of the performance in two scene datasets and Caltech-101 (%). (*)approximate value read from the graph.

Dataset	GLC + PLDA			Previous	
	L1	L2	L3	no SI	with SI
OT8	88.8	90.5	91.1	82.3 [19] 82.5 [3]	90.2 [19] 87.8 [3]
LSP15	80.0	83.2	84.1	72.7 [3] 74.8 [11]	83.7 [3] 81.4 [11]
Caltech-101	55.0	63.3	64.8	72.0* [1] 67.7 [3] 41.2 [11] 58.2 [7] 39.6 [8]	66.2 [20] 64.6 [11]



Figure 1: Sample images from the OT8 dataset.

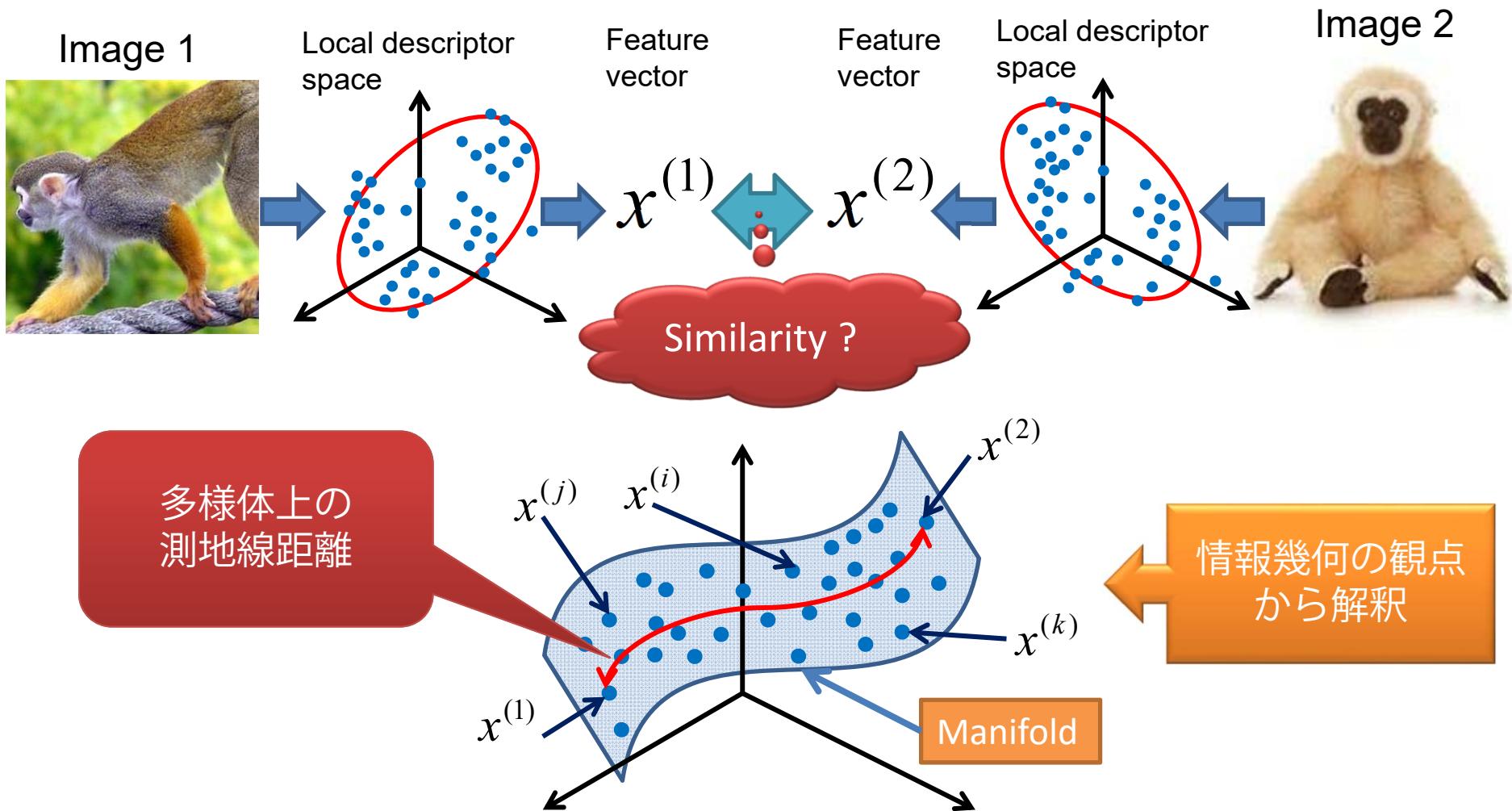


Figure 2: Additional seven classes in the LSP15 dataset.

Global Gaussian (GG)

H. Nakayama, T. Harada, and Y. Kuniyoshi. Global Gaussian Approach for Scene Categorization Using Information Geometry. In CVPR, 2010.

- 平均と分散を並べたGLCの表現は適切か？
- GLC間の距離計量は適切か？



Global Gaussian (GG)

H. Nakayama, T. Harada, and Y. Kuniyoshi. Global Gaussian Approach for Scene Categorization Using Information Geometry. In CVPR, 2010.

- 確率密度分布間の距離計量を正しく設定
- 情報幾何の考え方からGLCが自然に出てくる

η 座標系におけるsingle Gaussianの表現

$$\begin{aligned}\eta &= \sum_{1 \leq i \leq d} \eta_i e_i + \sum_{1 \leq i \leq j \leq d} \eta_{ij} e_{ij} \\ &= (\eta_1, \dots, \eta_d, \eta_{11}, \dots, \eta_{1d}, \eta_{22}, \dots, \eta_{2d}, \dots, \eta_{dd})^T \\ &= (\hat{\mu}_1, \dots, \hat{\mu}_d, \hat{\Sigma}_{11} + \hat{\mu}_1^2, \dots, \hat{\Sigma}_{1d} + \hat{\mu}_1 \hat{\mu}_d, \\ &\quad \hat{\Sigma}_{22} + \hat{\mu}_2^2, \dots, \hat{\Sigma}_{dd} + \hat{\mu}_d^2)^T.\end{aligned}$$

GLC

$$\mathbf{x}^{(j)} = \begin{pmatrix} \boldsymbol{\mu}^{(j)} \\ \text{upper}(R^{(j)}) \end{pmatrix}$$

GLCの厳密な距離

$$\begin{aligned}dist(\boldsymbol{\eta}(P), \boldsymbol{\eta}(Q)) &= \text{tr}(\Sigma_P \Sigma_Q^{-1}) + \text{tr}(\Sigma_Q \Sigma_P^{-1}) - 2d + \\ &\quad \text{tr}((\Sigma_P^{-1} + \Sigma_Q^{-1})(\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q)(\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q)^T)\end{aligned}$$

$$K_{kl}(P, Q) = \exp(-a \ dist(\boldsymbol{\eta}(P), \boldsymbol{\eta}(Q)))$$

Gauss分布間の
symmetric KL-
divergence

GLCの近似的な距離

Fisher Information Matrix

Linear-SVMにそのまま利用可

$$K_{ct}(P, Q) = \boldsymbol{\eta}(P)^T G^\eta(\boldsymbol{\eta}_c) \boldsymbol{\eta}(Q) \quad \Rightarrow \quad \zeta = (G^\eta(\boldsymbol{\eta}_c))^{1/2} \boldsymbol{\eta}$$

Global Gaussian (GG)

H. Nakayama, T. Harada, and Y. Kuniyoshi. Global Gaussian Approach for Scene Categorization Using Information Geometry. In CVPR, 2010.

- 性能評価

Table 5. Performances of global Gaussian, BoK, and combined approach (%). $L = 2$ spatial pyramid is implemented. Kernel PDA is used for classification. SURF descriptor is used for LSP15 and SIFT descriptor is used for 8-sports.

	LSP15	8-sports
GG (KL)	86.1±0.5	84.4±1.4
GG (ct-linear)	82.3±0.4	82.9±1.0
BoK200	81.1±0.7	79.6±1.1
BoK1000	82.5±0.7	81.5±1.7
GG (ct-linear) + BoK200	85.0±0.5	83.2±0.9
GG (ct-linear) + BoK1000	85.3±0.5	83.4±0.7

提案手法のスコア

Table 6. Performance comparison with previous work (%). For our method, $L = 2$ spatial pyramid is implemented, and kernel PDA is used for classification. We use the SURF descriptor for LSP15 and Indoor67, and the SIFT descriptor for 8-sports.

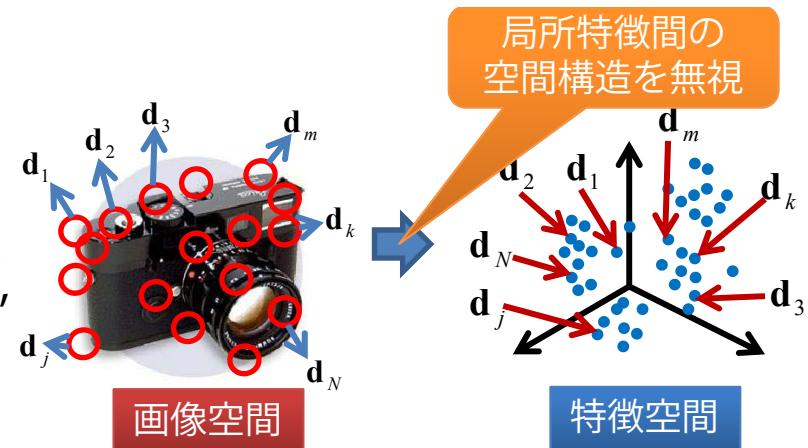
Method	LSP15	8-sports	Indoor67
GG (KL-div.)	86.1±0.5	84.4±1.4	45.5±1.1
GG (ct-linear) + BoK1000	85.3±0.5	83.4±0.7	44.9±1.3
Previous	85.2 [30] 84.1 [29] 83.7 [6]	84.2 [29] 73.4 [14]	25.0 [23]

従来手法のスコア

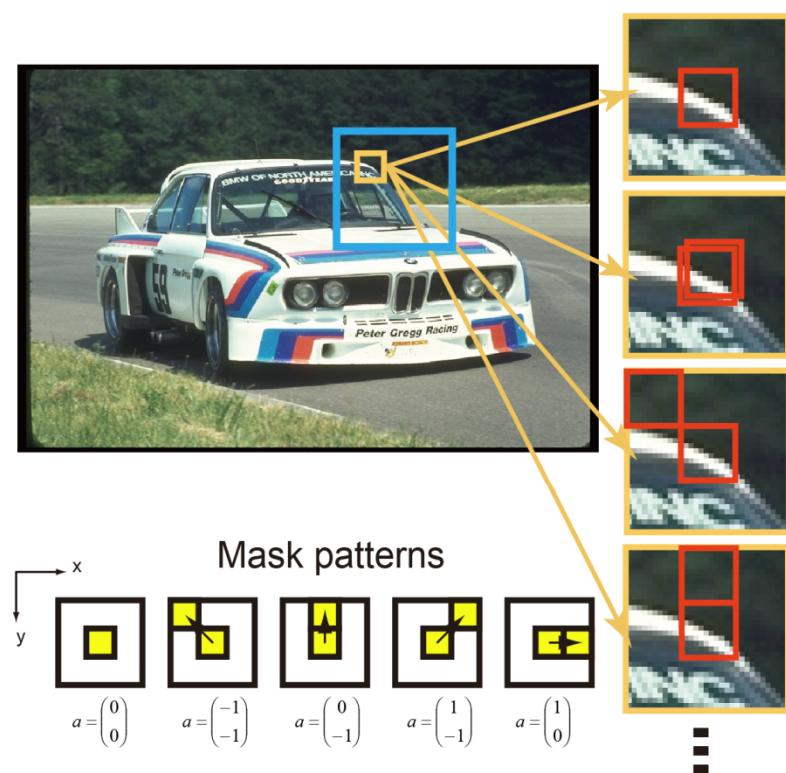
いずれのデータセットにおいても従来手法を上回る性能
平均, 共分散といったパラメータのみで計算可能！

局所空間情報の埋め込み

- T. Harada, H. Nakayama, and Y. Kuniyoshi. Improving Local Descriptors by Embedding Global and Local Spatial Information. In ECCV, 2010.
- HLACの一般化：任意の局所記述子の利用



Local Spatial Information Embedding



Local auto correlations

$$\frac{1}{N_J} \sum_{i \in J} \phi(\mathbf{r}_i) = \begin{pmatrix} m_1 \\ \vdots \\ m_d \end{pmatrix}$$

Sum of local auto correlations over the region

$$\frac{1}{N_J} \sum_{i \in J} \phi(\mathbf{r}_i) \phi(\mathbf{r}_i)^T = \begin{pmatrix} c_{11}^{(0)} & \cdots & c_{1d}^{(0)} \\ \vdots & \ddots & \vdots \\ c_{d1}^{(0)} & \cdots & c_{dd}^{(0)} \end{pmatrix}$$

Region feature

$$\mathbf{f}_j = \begin{pmatrix} m_1 \\ \vdots \\ m_d \\ c_{11}^{(0)} & \cdots & c_{1d}^{(0)} \\ \vdots & \ddots & \vdots \\ c_{d1}^{(0)} & \cdots & c_{dd}^{(0)} \\ \hline c_{11}^{(1)} & \cdots & c_{1d}^{(1)} \\ \vdots & \ddots & \vdots \\ c_{d1}^{(1)} & \cdots & c_{dd}^{(1)} \\ \hline c_{11}^{(2)} & \cdots & c_{1d}^{(2)} \\ \vdots & \ddots & \vdots \\ c_{d1}^{(2)} & \cdots & c_{dd}^{(2)} \end{pmatrix}$$

Local descriptor (SIFT, Ssim, ...)

$$\phi(\mathbf{r}_i) \phi(\mathbf{r}_i + \mathbf{a}_1)^T$$

$$\phi(\mathbf{r}_i) \phi(\mathbf{r}_i + \mathbf{a}_2)^T$$

$$\vdots$$

$$\phi(\mathbf{r}_i) \phi(\mathbf{r}_i + \mathbf{a}_N)^T$$

局所空間情報の埋め込み

T. Harada, H. Nakayama, and Y. Kuniyoshi. Improving Local Descriptors by Embedding Global and Local Spatial Information. In ECCV, 2010.

- 性能評価

Object classification: Caltech101 [Fei-Fei et al., 2004]

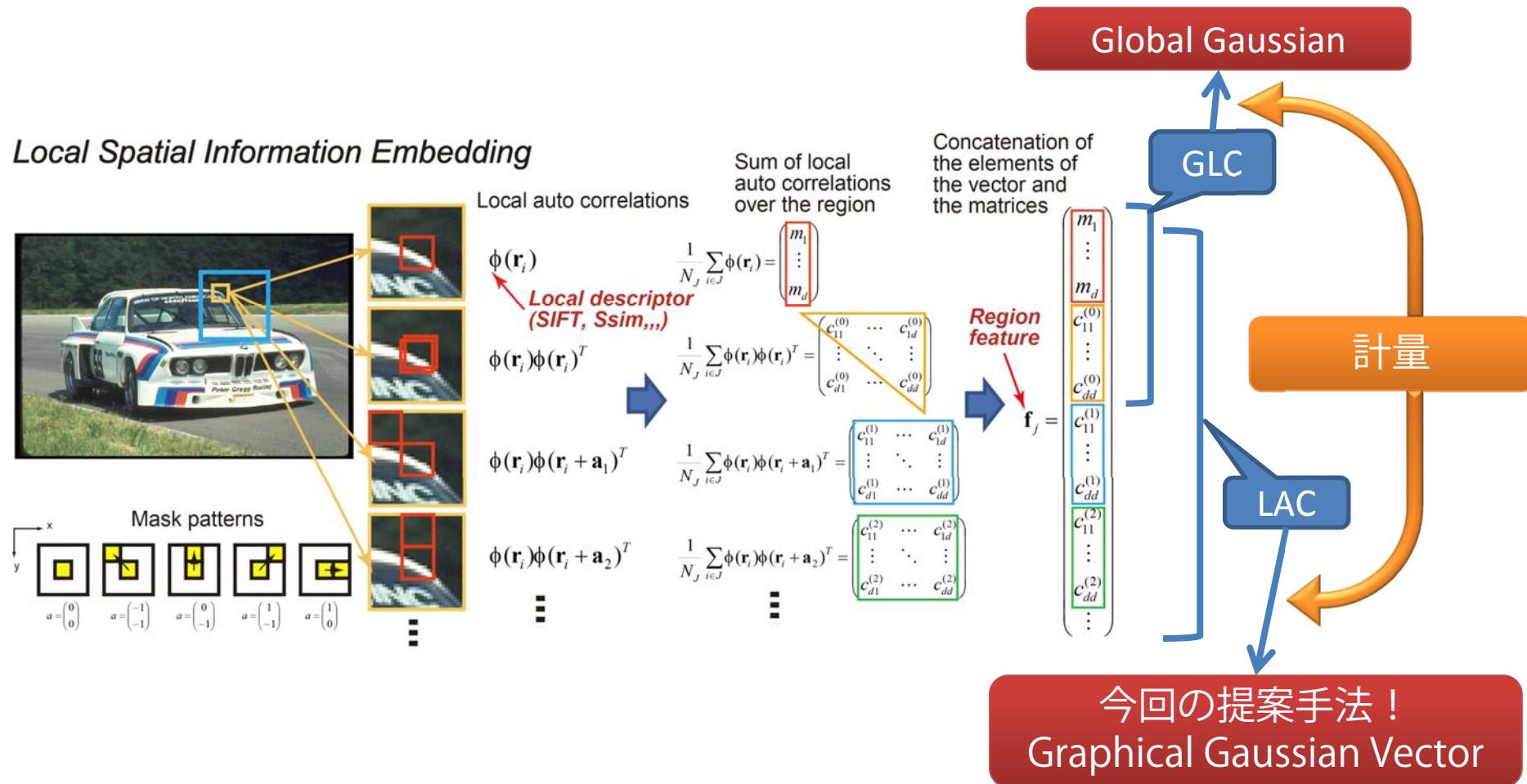
Classifier: Linear DA Training on 15 samples per category

Feature	Grid	LDA dim	Maps dim	PCA dim	Classification rate [%]	Learn [sec]	Classify [sec]
SIFT- (baseline)	2x2	32	4	32	28.1 ± 1.3	0.1	0.1
	4x4	101	16	120	42.9 ± 1.2	0.2	0.4
	6x6	101	36	120	44.8 ± 1.7	0.3	0.4
	8x8	101	64	120	44.4 ± 0.8	1.1	0.4
SIFT- +GLS (proposed)	2x2	101	4	600	46.3 ± 1.0	15.1	0.5
	4x4	101	4	600	50.2 ± 1.2	15.4	0.5
	6x6	101	4	600	52.6 ± 1.1	15.6	0.5
	8x8	101	4	600	53.4 ± 1.0	16	0.5
SS (baseline)	2x2	96	4	96	40.3 ± 1.4	0.1	0.1
	4x4	101	16	120	43.8 ± 1.4	0.5	0.4
	6x6	101	36	120	42.6 ± 1.5	5.3	0.5
	8x8	101	64	120	42.3 ± 1.3	2.4	0.4
SS+GLS (proposed)	2x2	101	4	350	52.6 ± 0.8	64	0.5
	4x4	101	6	350	52.5 ± 1.3	100	0.5

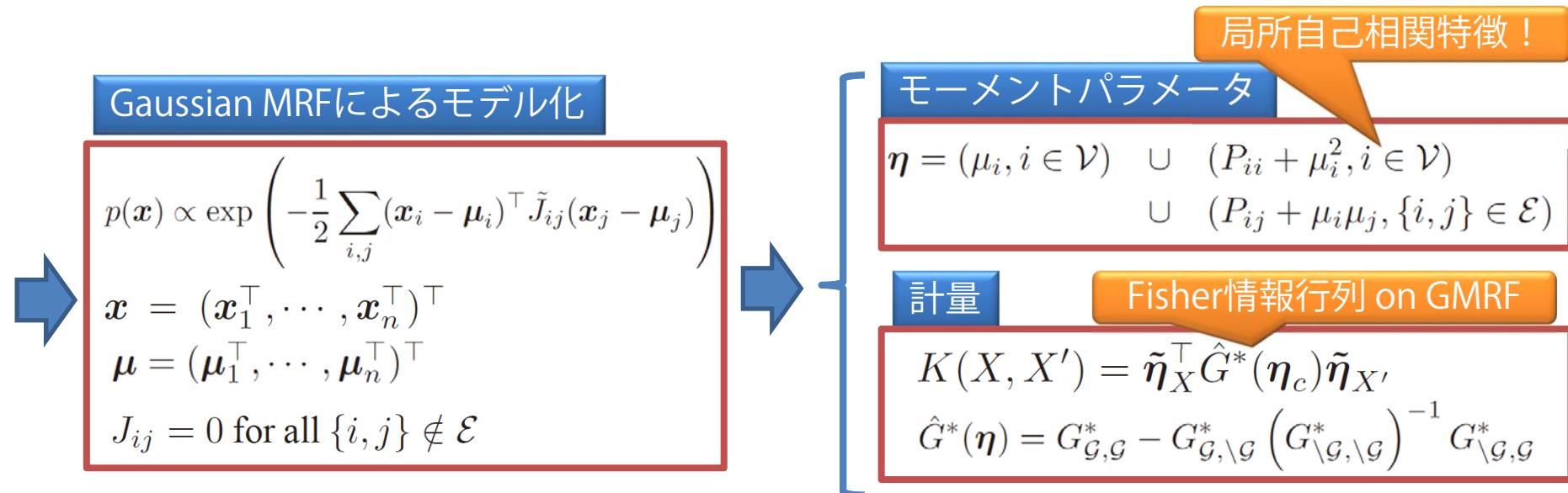
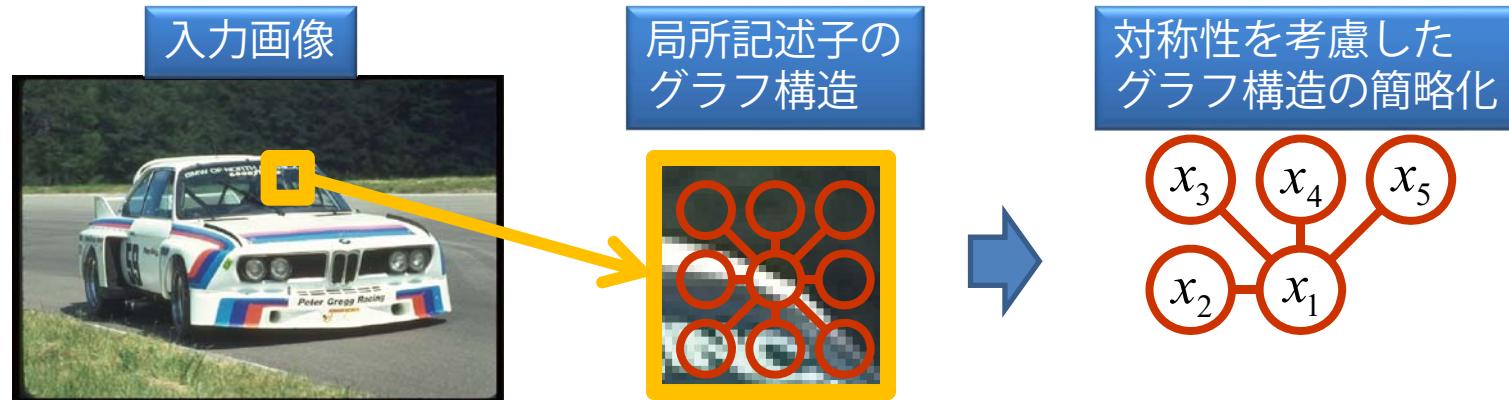
任意の局所記述子に利用可能であり、大幅な性能向上

局所自己相関特徴への計量の埋め込み

- GLCでは情報幾何の観点から適切な距離計量が定義できた。
- 局所記述子の局所自己相関特徴の適切な距離計量が決められないか？



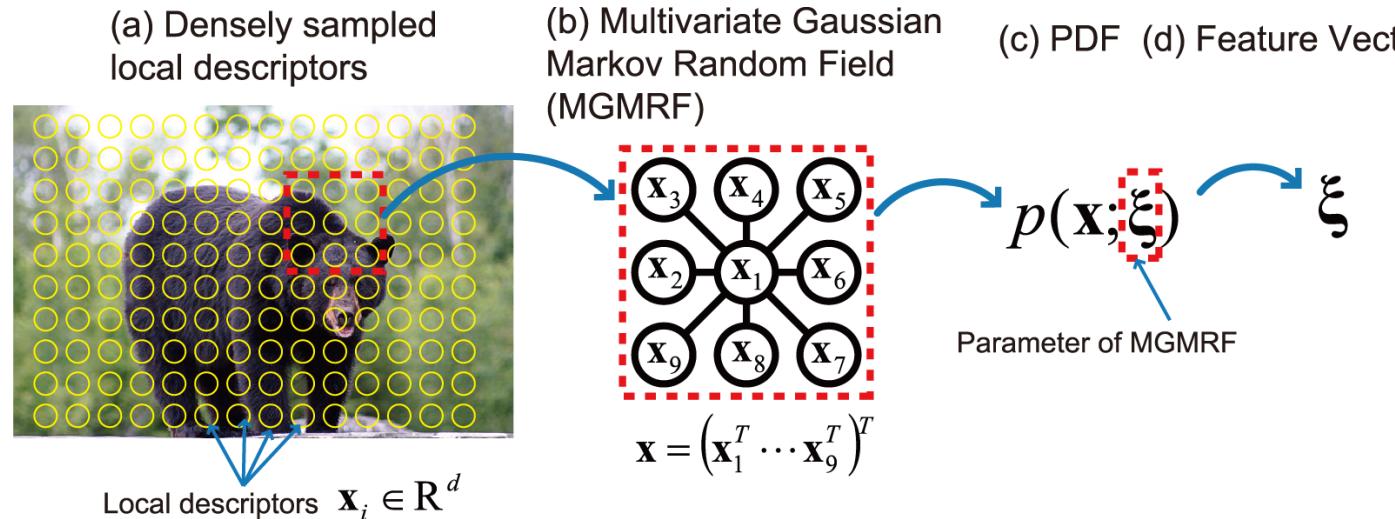
グラフィカルガウシアンベクトルの概要



グラフィカルガウシアンベクトル

$$\boldsymbol{\zeta} = \left(F_{\mathcal{G}, \mathcal{G}}^*(\boldsymbol{\eta}_c) - F_{\mathcal{G}, \setminus \mathcal{G}}^*(\boldsymbol{\eta}_c) \left(F_{\setminus \mathcal{G}, \setminus \mathcal{G}}^*(\boldsymbol{\eta}_c) \right)^{-1} F_{\setminus \mathcal{G}, \mathcal{G}}^*(\boldsymbol{\eta}_c) \right)^{1/2} \boldsymbol{\eta}.$$

局所特徴のGaussian MRFによるモデル化



Gaussian MRF

指指数型分布族 $p(\mathbf{x}; \boldsymbol{\theta}) = \exp \left(\boldsymbol{\theta}^\top \phi(\mathbf{x}) - \Phi(\boldsymbol{\theta}) + C(\mathbf{x}) \right)$

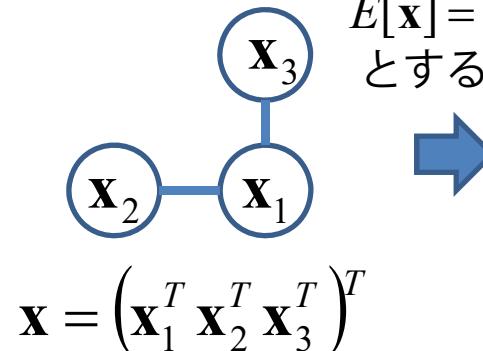
θ 座標系

自然パラメータ $\boldsymbol{\theta} = (h_i, i \in \mathcal{V}) \cup (-\frac{1}{2}J_{ii}, i \in \mathcal{V}) \cup (-J_{ij}, \{i, j\} \in \mathcal{E})$

η 座標系

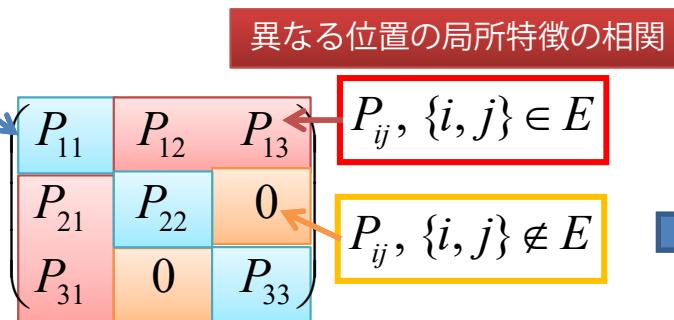
期待値パラメータ $\boldsymbol{\eta} = \mathbb{E}[\phi(\mathbf{x})] = (\mu_i, i \in \mathcal{V}) \cup (P_{ii} + \mu_i^2, i \in \mathcal{V}) \cup (P_{ij} + \mu_i \mu_j, \{i, j\} \in \mathcal{E})$

【例】



$P_{ii}, i \in V$

$P = E[\mathbf{x}\mathbf{x}^T]$



$P_{ij} = E[\mathbf{x}_i \mathbf{x}_j^T]$

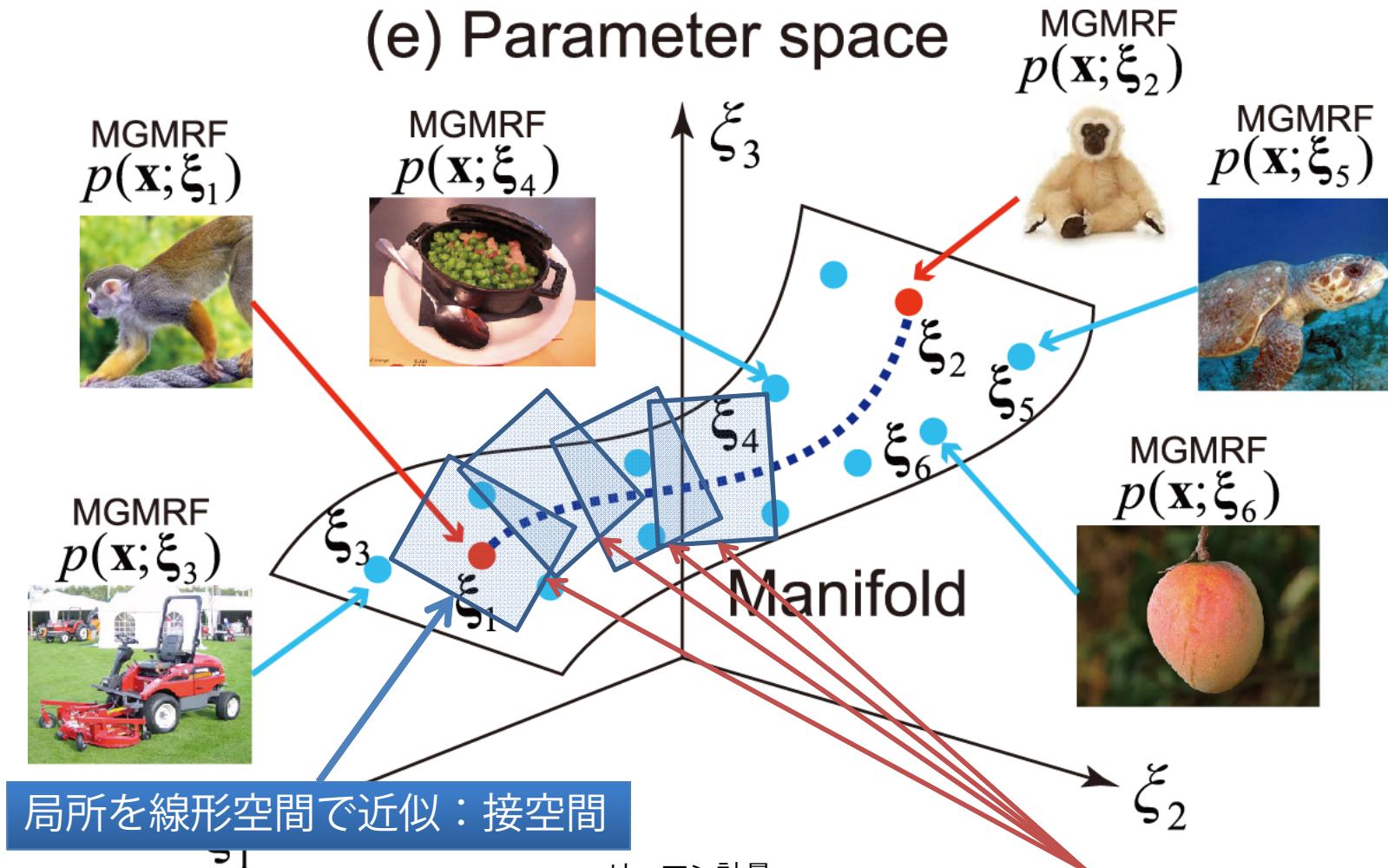
2と3の間にはエッジ
はないので相関0

$\boldsymbol{\eta} = \begin{pmatrix} \text{upper}[P_{11}] \\ \text{upper}[P_{22}] \\ \text{upper}[P_{33}] \\ \text{vec}[P_{12}] \\ \text{vec}[P_{13}] \end{pmatrix}$

局所自己相関
特徴の一般形

空間の構造

η 座標系でいいのか？



$$ds^2 = \text{KL}[p(x; \eta) : p(x; \eta + d\eta)] = \frac{1}{2} d\eta^\top G^* d\eta$$

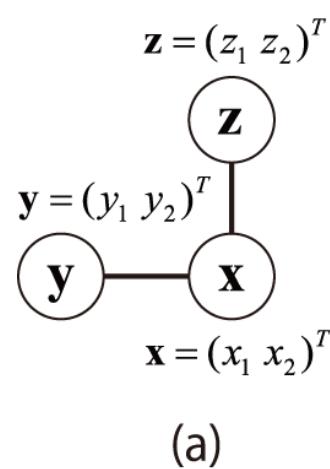
リーマン計量

接空間での計量：フィッシャー情報行列

指指数型分布族において η 座標ではユークリッド空間のような平坦な空間を実現できる (-1-平坦) .
ただし計量は場所により異なる。

Gaussian MRFにおけるFisher情報行列

- すべてのノードが接続されたグラフ構造におけるFisher情報行列を計算
 - Full GaussianのFisher情報行列の計算



$$F^* = \begin{array}{|c|c|c|c|} \hline & \begin{matrix} x_1, x_2, \dots, z_1, z_2 \\ x_1 x_2, x_2 x_2, \dots, z_1 z_1, z_2 z_2 \\ \vdots \\ z_1 z_2 \end{matrix} & \begin{matrix} x_1 x_1, x_2 x_2, \dots, z_1 z_1, z_2 z_2 \\ x_1 y_2, y_1 x_2, z_1 z_2 \\ \vdots \\ x_1 y_2, y_1 x_2, z_1 z_2 \end{matrix} & \begin{matrix} x_1 x_2, y_1 y_2, z_1 z_2 \\ x_1 y_1, x_1 y_2, x_2 y_1, x_2 y_2, \dots, z_1 z_1, z_1 z_2, z_2 z_1, z_2 z_2 \end{matrix} \\ \hline \begin{matrix} x_1 \\ x_2 \\ \vdots \\ z_1 \\ z_2 \end{matrix} & F_{ij}^* & F_{i,pp}^* & F_{i,pq}^* \\ \hline \begin{matrix} x_1 x_1 \\ x_2 x_2 \\ \vdots \\ z_1 z_1 \\ z_2 z_2 \end{matrix} & F_{i,pp}^* & F_{pp,rr}^* & F_{rr,pq}^* \\ \hline \begin{matrix} x_1 x_2 \\ y_1 y_2 \\ z_1 z_2 \\ \vdots \\ y_1 z_1 \\ y_2 z_2 \\ z_1 z_2 \end{matrix} & F_{i,pq}^* & F_{rr,pq}^* & F_{pq,rs}^* \\ \hline \begin{matrix} x_1 y_1 \\ x_1 y_2 \\ x_2 y_1 \\ x_2 y_2 \\ \vdots \\ y_1 z_1 \\ y_2 z_2 \\ z_1 z_2 \end{matrix} & F_{\backslash GG}^* & F_{\backslash GG}^* & F_{\backslash GG}^* \\ \hline \end{array}$$

(b)

$$G^* = \begin{array}{|c|c|c|c|} \hline & \text{Yellow} & \text{Blue} & \text{Yellow} \\ \hline \text{Yellow} & F_{GG}^* & & \\ \hline \text{Blue} & & & \\ \hline \text{Yellow} & & & \\ \hline \end{array} - \begin{array}{|c|c|c|c|} \hline & \text{Yellow} & \text{Blue} & \text{Yellow} \\ \hline \text{Yellow} & F_{G\backslash G}^* & \times & F_{\backslash G\backslash G}^* \times \begin{array}{|c|c|c|c|} \hline & \text{Yellow} & \text{Blue} & \text{Yellow} \\ \hline \text{Yellow} & & & \\ \hline \text{Blue} & & & \\ \hline \text{Yellow} & & & \\ \hline \end{array}^{-1} & F_{\backslash GG}^* \\ \hline \end{array}$$

(c)

グラフィカルガウシアンベクトル のアルゴリズム

Algorithm 1 Calculation of GGV.

Input: An image region J , and the Fisher information matrix F^*

Output: GGV ζ

1. Calculate zeroth- and first-order local auto-correlations of local features:

$$\bar{\mathbf{x}} = \frac{1}{N_J} \sum_{j \in J} \mathbf{x}(\mathbf{r}_j), \quad C(\mathbf{a}_i) = \frac{1}{N_J} \sum_{j \in J} \mathbf{x}(\mathbf{r}_j) \mathbf{x}(\mathbf{r}_j + \mathbf{a}_i)^\top$$

2. Estimate the expectation parameters:

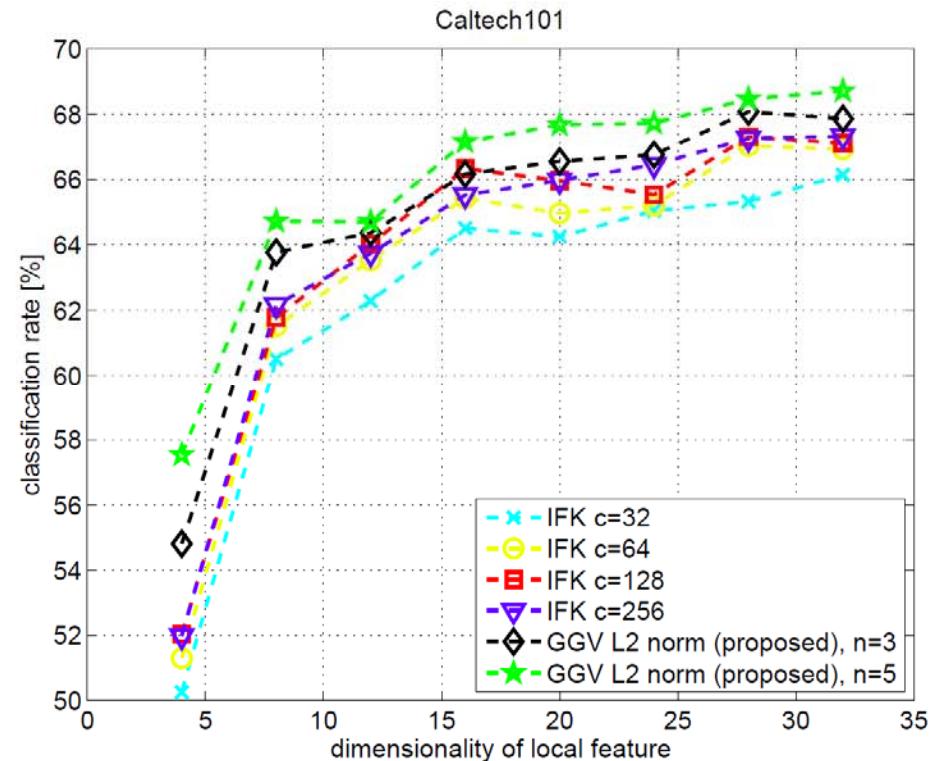
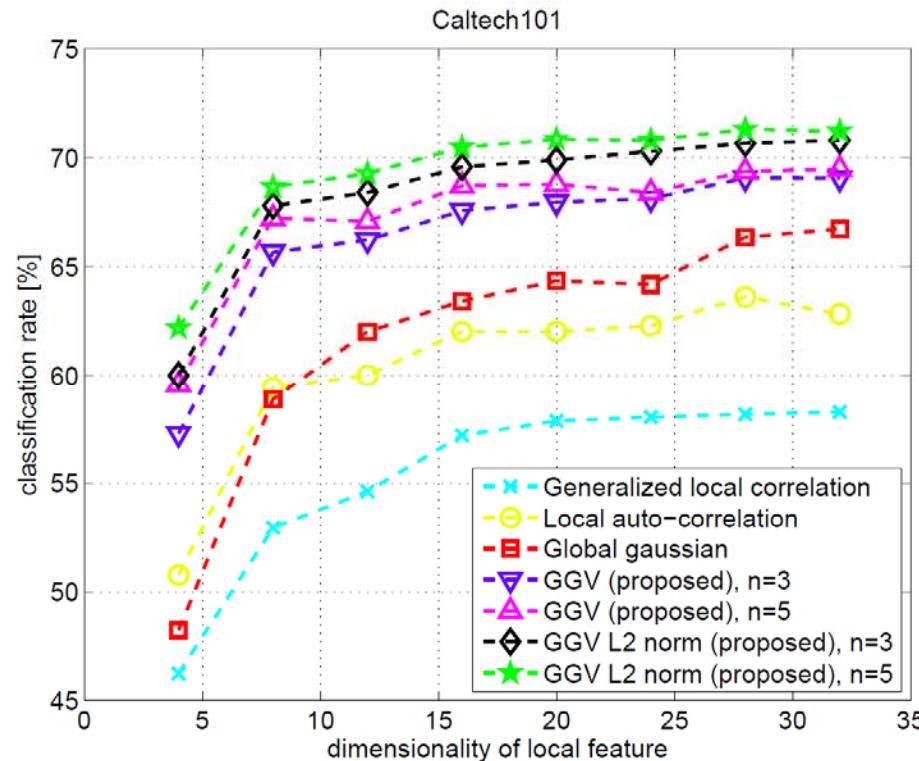
$$\boldsymbol{\eta} = (\bar{\mathbf{x}}^\top \cdots \bar{\mathbf{x}}^\top f^\top(C(\mathbf{0})) \cdots f^\top(C(\mathbf{0})) g^\top(C(\mathbf{a}_1)) \cdots g^\top(C(\mathbf{a}_{n-1})))^\top.$$

3. Embed the Riemannian metric into the expectation parameters:

$$\zeta = \left(F_{\mathcal{G}, \mathcal{G}}^*(\boldsymbol{\eta}_c) - F_{\mathcal{G}, \setminus \mathcal{G}}^*(\boldsymbol{\eta}_c) (F_{\setminus \mathcal{G}, \setminus \mathcal{G}}^*(\boldsymbol{\eta}_c))^{-1} F_{\setminus \mathcal{G}, \mathcal{G}}^*(\boldsymbol{\eta}_c) \right)^{1/2} \boldsymbol{\eta}$$

Graphical Gaussian Vector (GGV)

- 実験結果



いずれのデータセットにおいてもGlobal Gaussianを上回る性能
平均, 共分散といったパラメータのみで計算可能！→高効率な計算
最新手法と同等の性能！

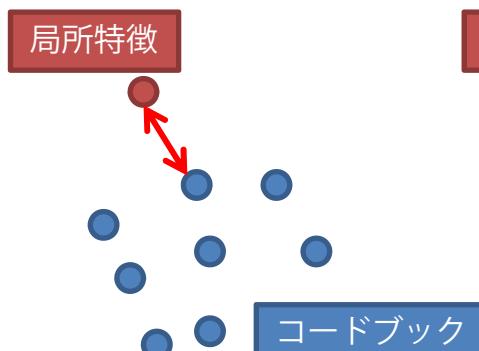
多様体によるコーディング

スパース符号化の比較

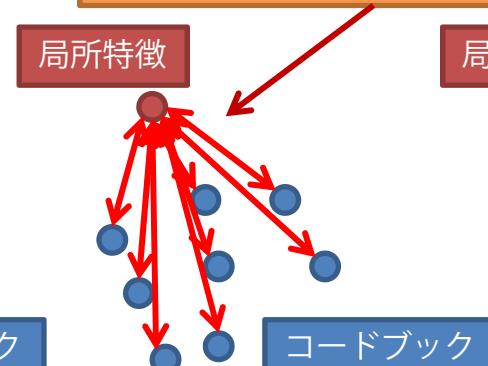
- BoF
 - 局所特徴が一つのコードワードに割り当てられる
- BoFのGMMによる表現
 - 局所特徴が全てのコードワードと関係を持つ
- スパース符号化
 - 局所特徴が少數のコードワードと関係を持つ
- 局所線形制約符号化
 - 局所特徴が局所の少數コードワードと関係を持つ



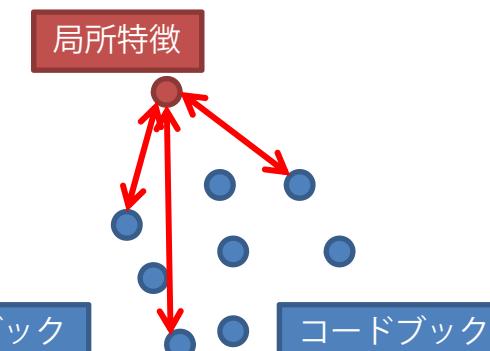
(注) コードワードへ割り付け
る確率なので他と意味が違う



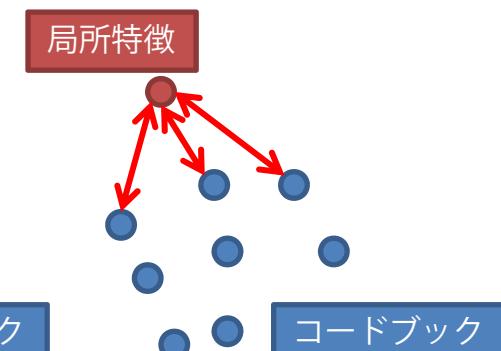
(a) BoF



(b) GMM



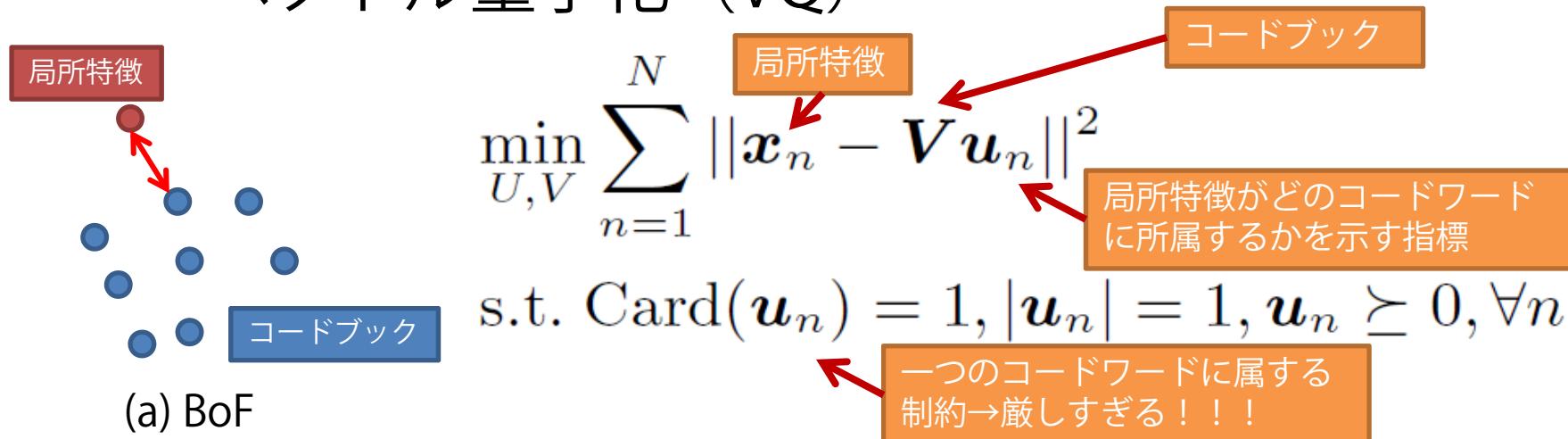
(c) Sparse Coding



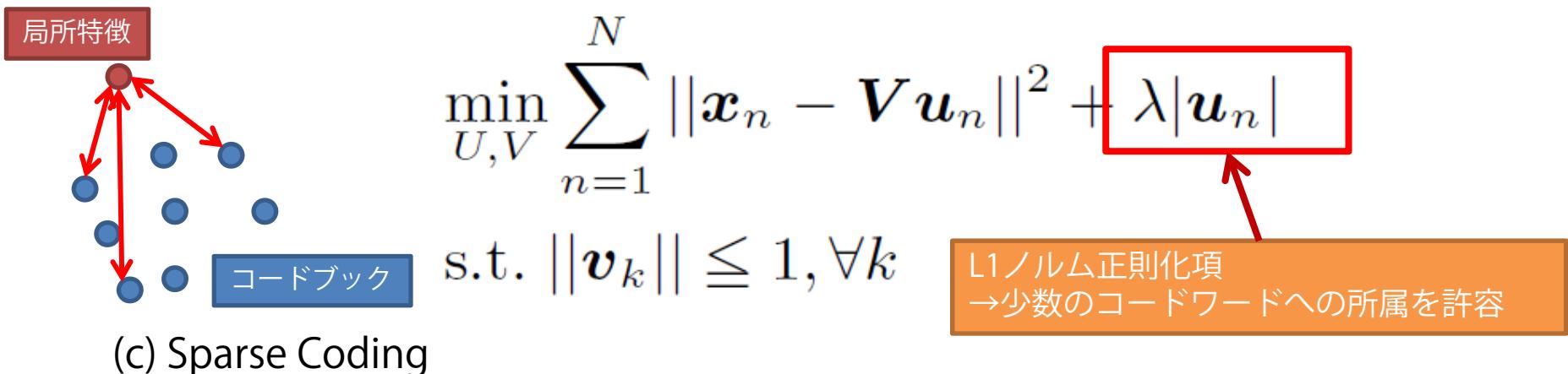
(d) LCC

スパース符号化の定式化

- Bag of Visual Words
 - ベクトル量子化 (VQ)



- スパース符号化 (Sparse Coding)

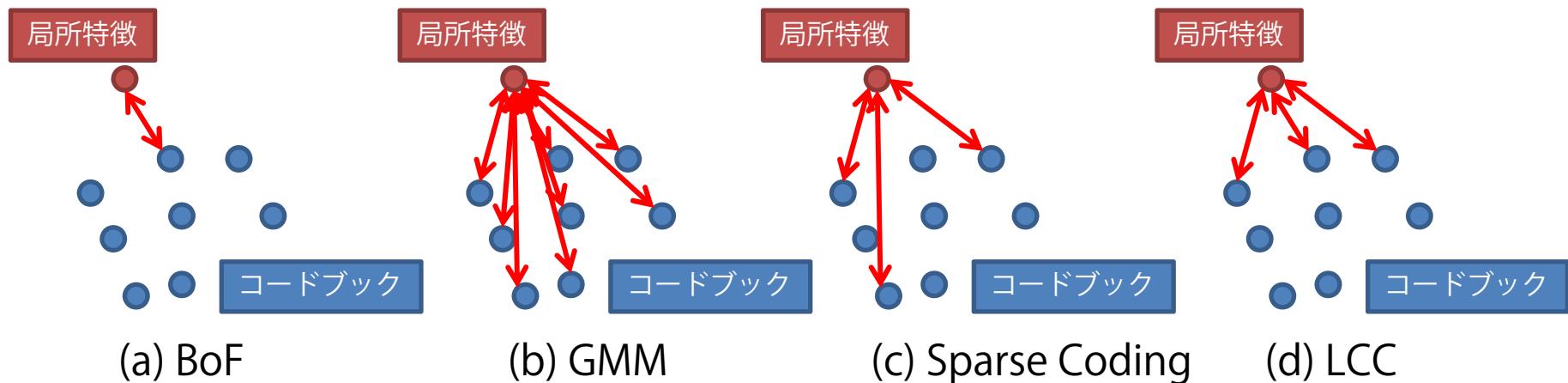


L1正則化の役割

- コードブックは局所特徴の次元数よりも多く、過剰 ($K > D$) なため、under determinedな系である。つまり情報が不足して解を定められない状況にある。そのためL1正則化により解を定めることができるとなる。
- スパース性の事前知識を用いることによって局所特徴の顕著なパターンを捉えることができる。
- ベクトル量子化よりもスパース符号化の方が量子化誤差を低減させられる。

局所線形制約符号化と他符号化の比較

- BoF
 - 局所特徴が一つのコードワードに割り当てられる
- BoFのGMMによる表現
 - 局所特徴が全てのコードワードと関係を持つ
- スパース符号化
 - 局所特徴が少数のコードワードと関係を持つ
- 局所線形制約符号化
 - 局所特徴が局所の少数コードワードと関係を持つ



局所座標符号化

Local Coordinate Coding (LCC)

- K. Yu, T. Zhang, and Y. Gong. Nonlinear learning using local coordinate coding. NIPS, 2009.

http://www.image-net.org/challenges/LSVRC/2010/ILSVRC2010_NEU-UIUC.pdf

$$\mathbf{X}(d \times N) \approx \mathbf{B}(d \times D) \times \mathbf{Z}(D \times N)$$

Assume \mathbf{B} is given.

Sparse coding:

$$\mathbf{z}^* = \arg \min_{\mathbf{z}} \frac{1}{2} \|\mathbf{x} - \mathbf{Bz}\|^2 + \lambda \sum_{i=1}^D |z_i|$$

LCC: K. Yu et. al, NIPS 2009

局所性がスパースネスよりも本質！！

$$\mathbf{z}^* = \arg \min_{\mathbf{z}} \frac{1}{2} \|\mathbf{x} - \mathbf{Bz}\|^2 + \lambda \sum_{i=1}^D \|\mathbf{x} - \mathbf{b}_i\|^2 |z_i|$$

Explicitly enforcing locality constraint

なぜ局所座標符号化が良いのか？

http://www.image-net.org/challenges/LSVRC/2010/ILSVRC2010_NEU-UIUC.pdf

$$f(\mathbf{x}) \approx \sum_{i=1}^D z_i(\mathbf{x}) w_i$$

e.g. nonlinear separating hyperplane

$$\begin{aligned} & |f(\mathbf{x}) - \sum_{i=1}^D z_i(\mathbf{x}) f(\mathbf{b}_i)| \leftarrow \text{Functional approximation error} \\ & \leq \alpha \underbrace{\|\mathbf{x} - \mathbf{Bz}(\mathbf{x})\|}_{\text{Coding error}} + \beta \underbrace{\sum_{i=1}^D \|\mathbf{x} - \mathbf{b}_i\|^2 |z_i(\mathbf{x})|}_{\text{Locality term}} \end{aligned}$$

- よりよく近似するためには
 - 局所特徴に対して局所性を有すること
 - 局所特徴の再構築誤差を減らすこと

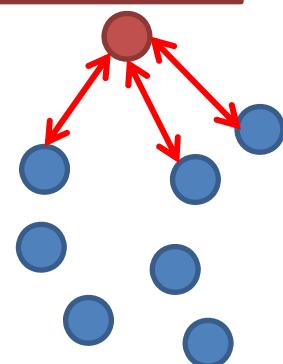
局所制約線形符号化

- 局所制約線形符号化
 - Locality-constrained Linear Coding (LLC)
 - J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. CVPR, 2010.
 - 局所座標符号化Local Coordinate Coding (LCC)の高速な実装
 - K. Yu, T. Zhang, and Y. Gong. Nonlinear learning using local coordinate coding. NIPS, 2009.

1) 局所特徴のK近傍のコード
ワードを探索

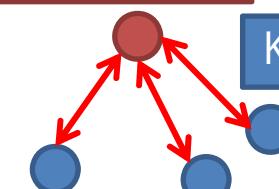
2) 局所特徴をK近傍コードワー
ドを用いて再構築

局所特徴



コードブック

局所特徴

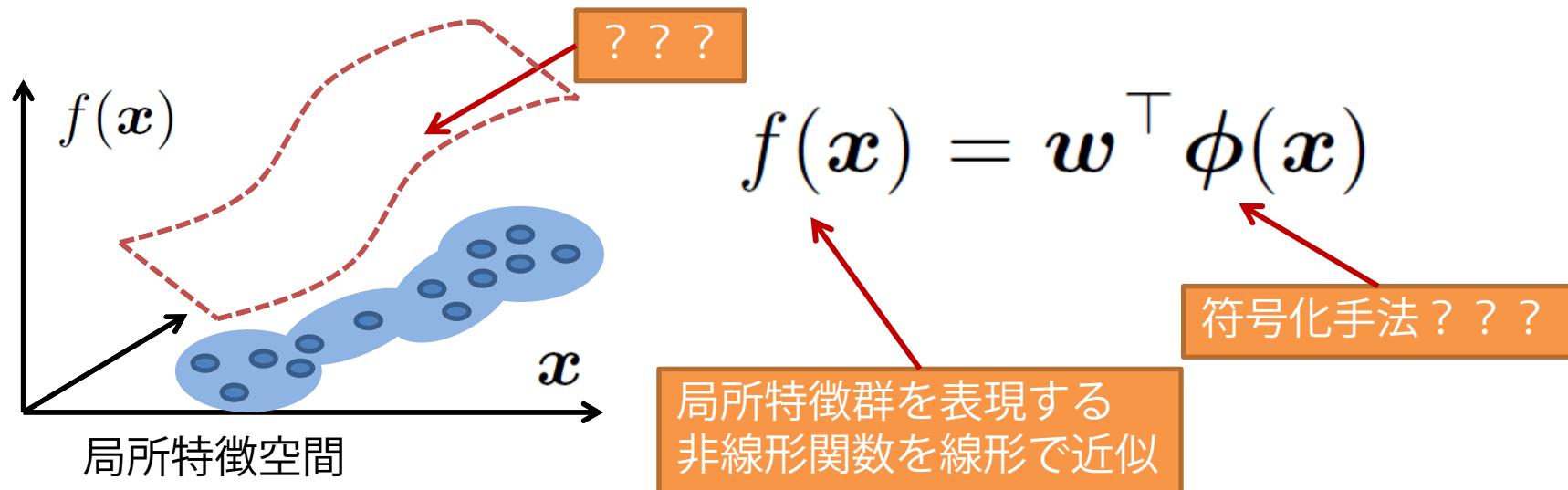


$$\begin{aligned} & \min_{\tilde{U}} \sum_{n=1}^N \| \mathbf{x}_n - \mathbf{V}_n \tilde{\mathbf{u}}_n \|^2 \\ \text{s.t. } & \mathbf{1}^\top \tilde{\mathbf{u}}_n = 1, \forall n \end{aligned}$$

局所線形埋込み (Local Linear Embedding, LLE) と比較して,
局所制約線形符号化はコードブックの学習が入る点で異なる。

スーパーベクトル符号化 Super-Vector Coding

- X. Zhou, K. Yu, T. Zhang, and T.S. Huang. Image classification using super-vector coding of local image descriptors. ECCV, 2010.
- BoF や混合ガウス分布を用いたBoF の改善手法
 - 特徴空間における局所特徴の分布の表現を得るプロセスと解釈できた.
- ここでも高次元空間における局所特徴分布を表現する, なめらかな非線形関数 $f(x)$ の学習について考える.
- 非線形関数 $f(x)$ を線形表現可能な符号化手法 $\phi(x)$ を求める.



スーパーベクトル符号化の導出

- 局所特徴をコードブックを利用して近似

$$\mathbf{x} \approx \sum_{k=1}^K \gamma_x(k) \mathbf{v}_k$$

負担率のようなもの コードワードk

$\gamma_x = [\gamma_x(1), \dots, \gamma_x(K)], \sum_{k=1}^K \gamma_x(k) = 1$

- β Lipschitz derivative smooth

$$|f(\mathbf{x}) - f(\mathbf{x}') - \nabla f(\mathbf{x}')^\top (\mathbf{x} - \mathbf{x}')| \leq \frac{\beta}{2} \|\mathbf{x} - \mathbf{x}'\|^2$$

コードワードの代入

$$\downarrow \quad \mathbf{x}' = \mathbf{v}^x$$

$$|f(\mathbf{x}) - f(\mathbf{v}^x) - \nabla f(\mathbf{v}^x)^\top (\mathbf{x} - \mathbf{v}^x)| \leq \frac{\beta}{2} \|\mathbf{x} - \mathbf{v}^x\|^2$$

$$f(\mathbf{x}) = f(\mathbf{v}^x) + \nabla f(\mathbf{v}^x)^\top (\mathbf{x} - \mathbf{v}^x) \cdots (\star)$$

関数f(x)の1次近似のUpper boundに関する式

$\|\mathbf{x}-\mathbf{v}\|$ が小さければ近似精度が向上

- スーパーベクトル符号化

$$f(\mathbf{x}) \approx \mathbf{w}^\top \phi(\mathbf{x})$$

式(☆)を分解！

Super Vector Coding

$$\phi(\mathbf{x}) = [s\gamma_x(k), \gamma_x(k)(\mathbf{x} - \mathbf{v}_k)^\top]^\top_{v_k \in \mathcal{V}}$$

$$\mathbf{w} = \left[\frac{1}{s} f(\mathbf{v}_k), (\nabla f(\mathbf{v}_k))^\top \right]^\top_{v_k \in \mathcal{V}}$$

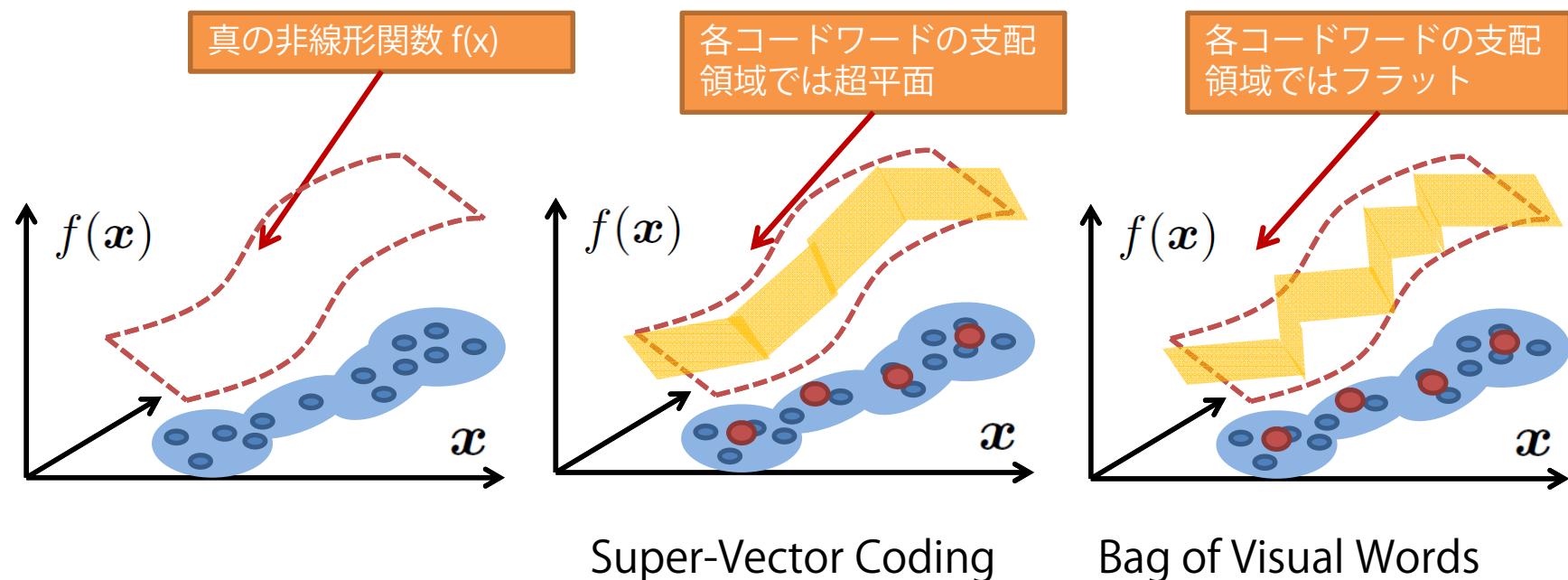
スーパーべクトル符号化の解釈

- スーパーべクトル符号化の例
 - コードワード数 : 3, $\gamma = [0 \ 1 \ 0]'$

Super Vector Coding

$$\phi(\mathbf{x}) = \left[s\gamma_x(k), \gamma_x(k)(\mathbf{x} - \mathbf{v}_k)^\top \right]_{v_k \in \mathcal{V}}^\top \quad \Rightarrow \quad \phi(\mathbf{x}) = \begin{bmatrix} 0, \dots, 0, \underbrace{s, (\mathbf{x} - \mathbf{v})^\top}_{d+1 \text{ dim}}, 0, \dots, 0 \end{bmatrix}^T$$

- スーパーべクトル符号化とBoF



スーパーべクトル符号化とフィッシューベクトル

・ フィッシューベクトル

$$\begin{aligned}\frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \pi_k} &= \sum_{n=1}^N \left[\frac{\gamma_n(k)}{\pi_k} - \frac{\gamma_n(1)}{\pi_1} \right] \\ \frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k^d} &= \sum_{n=1}^N \gamma_n(k) \left[\frac{\mathbf{x}_n^d - \boldsymbol{\mu}_k^d}{(\sigma_k^d)^2} \right] \\ \frac{\partial \mathcal{L}(\mathcal{X}|\boldsymbol{\theta})}{\partial \sigma_k^d} &= \sum_{n=1}^N \gamma_n(k) \left[\frac{(\mathbf{x}_n^d - \boldsymbol{\mu}_k^d)^2}{(\sigma_k^d)^3} - \frac{1}{\sigma_k^d} \right]\end{aligned}$$

GMMのBoFとほぼ同じ

局所特徴 \mathbf{x}_n とGMMの各コンポーネント k の平均との差分

・ スーパーベクトル符号化

$$\phi(\mathbf{x}) = \left[s\gamma_x(k), \gamma_x(k)(\mathbf{x} - \mathbf{v}_k)^\top \right]^\top_{v_k \in \mathcal{V}}$$

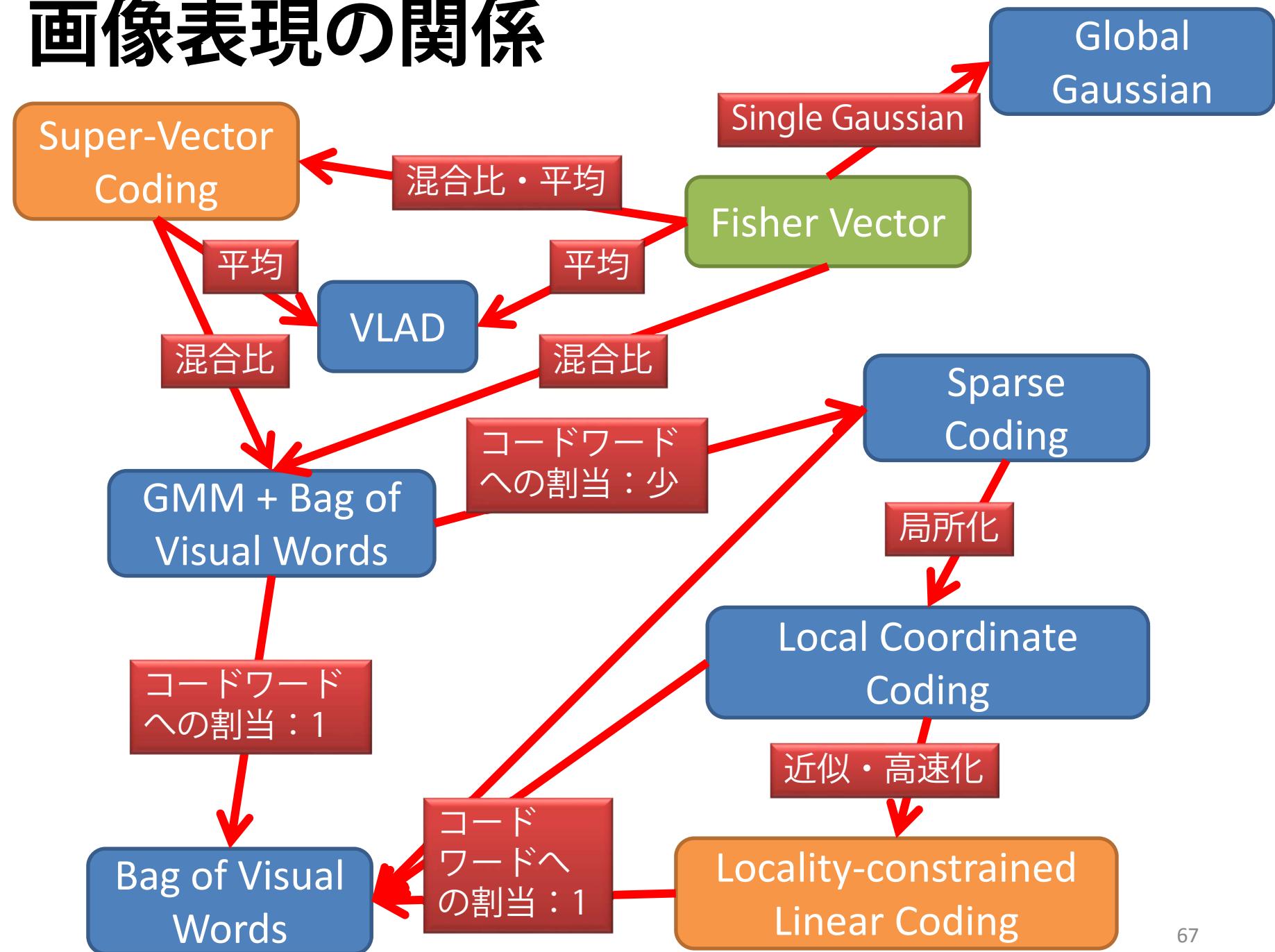
負担率

局所特徴 \mathbf{x}_n とコードワードとの差分

- 混合比：一定
- 分散：一定

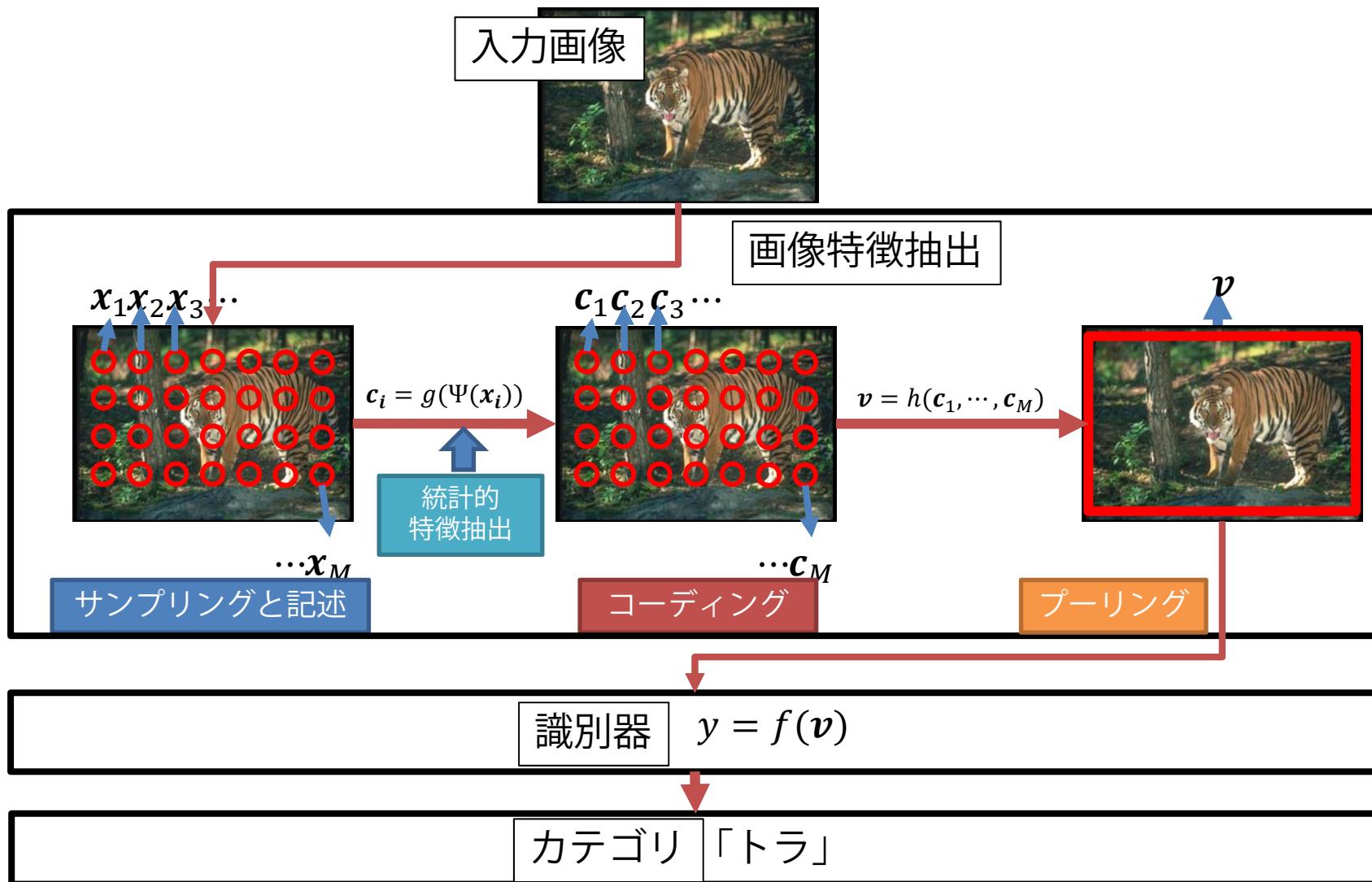
→スーパーべクトル符号化は
フィッシューベクトルの混合比
と平均に関する要素と同じ
• (注) 分散を考えていないので
フィッシューとは言い難い。

画像表現の関係



プリング

カテゴリ認識の流れ



Spatial Pyramid Representation

S. Lazebnik, C. Schmid, and J. Ponce

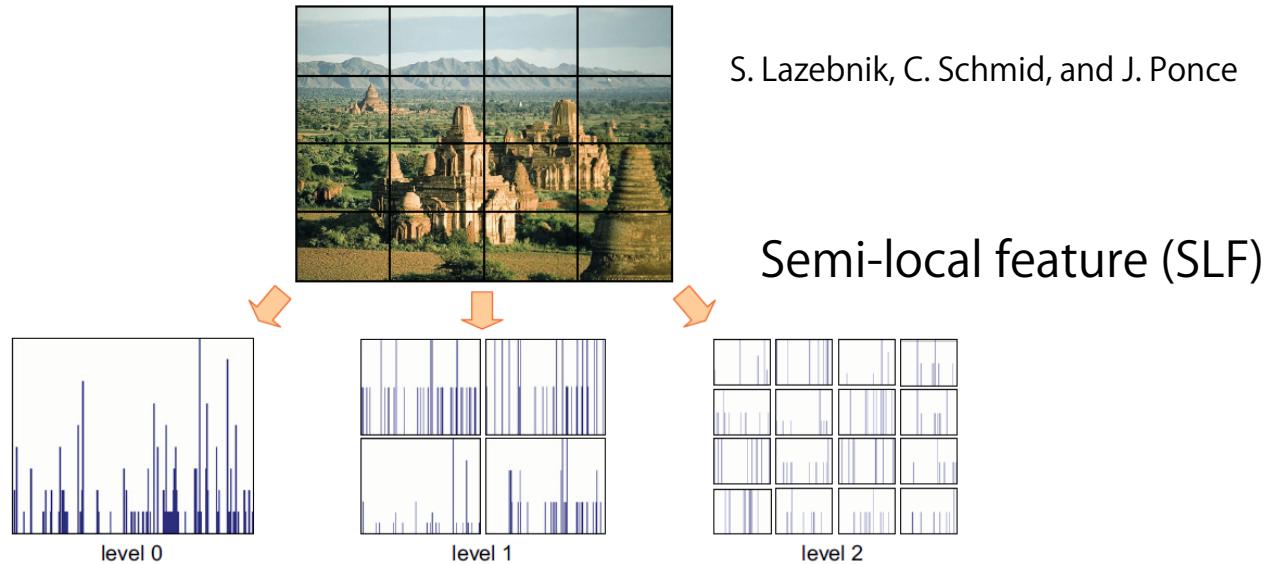


Fig. 1.1. A schematic illustration of the spatial pyramid representation. A spatial pyramid is a collection of orderless feature histograms computed over cells defined by a multi-level recursive image decomposition. At level 0, the decomposition consists of just a single cell, and the representation is equivalent to a standard bag of features. At level 1, the image is subdivided into four quadrants, yielding four feature histograms, and so on. Spatial pyramids can be matched using the *pyramid kernel*, which weights features at higher levels more highly, reflecting the fact that higher levels localize the features more precisely (see Section 1.2).

- Level0: Global featureと同じ
- Level1: 2x2のcellに分割し各cellでSLFを計算
- Level2: 4x4のcellに分割し各cellでSLFを計算

スパース符号化空間ピラミッド

- 空間ピラミッド
 - 符号化された局所特徴群 U から一つの特徴ベクトル f を得る手段

- プーリング (pooling)

$$f = \mathcal{F}(U)$$

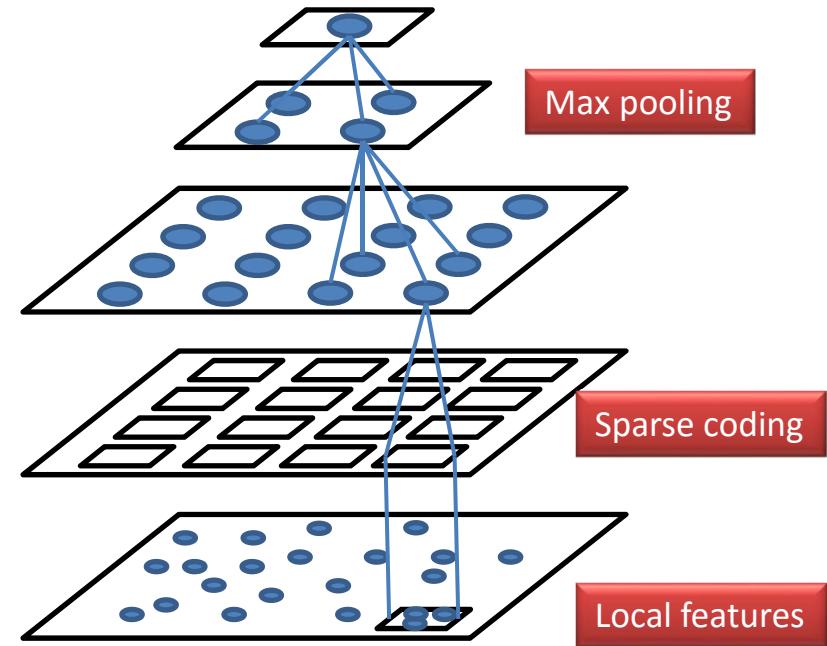
- 平均プーリング
average pooling

$$f = \frac{1}{N} \sum_{n=1}^N u_n$$

BoFはこれを利用

- 最大値プーリング
max pooling

$$f^d = \max\{|u_1^d|, |u_2^d|, \dots, |u_N^d|\}$$



J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. CVPR, 2009.

最大値プーリングの効果

- Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. CVPR, 2010.

Method	Caltech-101, 30 training examples		15 Scenes, 100 training examples	
	Average Pool	Max Pool	Average Pool	Max Pool
Results with basic features, SIFT extracted each 8 pixels				
Hard quantization, linear kernel	51.4 \pm 0.9 [256]	64.3 \pm 0.9 [256]	73.9 \pm 0.9 [1024]	80.1 \pm 0.6 [1024]
Hard quantization, intersection kernel	64.2 \pm 1.0 [256] (1)	64.3 \pm 0.9 [256]	80.8 \pm 0.4 [256] (1)	80.1 \pm 0.6 [1024]
Soft quantization, linear kernel	57.9 \pm 1.5 [1024]	69.0 \pm 0.8 [256]	75.6 \pm 0.5 [1024]	81.4 \pm 0.6 [1024]
Soft quantization, intersection kernel	66.1 \pm 1.2 [512] (2)	70.6 \pm 1.0 [1024]	81.2 \pm 0.4 [1024] (2)	83.0 \pm 0.7 [1024]
Sparse codes, linear kernel	61.3 \pm 1.3 [1024]	71.5 \pm 1.1 [1024] (3)	76.9 \pm 0.6 [1024]	83.1 \pm 0.6 [1024] (3)
Sparse codes, intersection kernel	70.3 \pm 1.3 [1024]	71.8 \pm 1.0 [1024] (4)	83.2 \pm 0.4 [1024]	84.1 \pm 0.5 [1024] (4)
Results with macrofeatures and denser SIFT sampling				
Hard quantization, linear kernel	55.6 \pm 1.6 [256]	70.9 \pm 1.0 [1024]	74.0 \pm 0.5 [1024]	80.1 \pm 0.5 [1024]
Hard quantization, intersection kernel	68.8 \pm 1.4 [512]	70.9 \pm 1.0 [1024]	81.0 \pm 0.5 [1024]	80.1 \pm 0.5 [1024]
Soft quantization, linear kernel	61.6 \pm 1.6 [1024]	71.5 \pm 1.0 [1024]	76.4 \pm 0.7 [1024]	81.5 \pm 0.4 [1024]
Soft quantization, intersection kernel	70.1 \pm 1.3 [1024]	73.2 \pm 1.0 [1024]	81.8 \pm 0.4 [1024]	83.0 \pm 0.4 [1024]
Sparse codes, linear kernel	65.7 \pm 1.4 [1024]	75.1 \pm 0.9 [1024]	78.2 \pm 0.7 [1024]	83.6 \pm 0.4 [1024]
Sparse codes, intersection kernel	73.7 \pm 1.3 [1024]	75.7 \pm 1.1 [1024]	83.5 \pm 0.4 [1024]	84.3 \pm 0.5 [1024]

Table 1. Average recognition rate on Caltech-101 and 15-Scenes benchmarks, for various combinations of coding, pooling, and classifier types. The codebook size shown inside brackets is the one that gives the best results among 256, 512 and 1024. Linear and histogram intersection kernels are identical when using hard quantization with max pooling (since taking the minimum or the product is the same for binary vectors), but results have been included for both to preserve the symmetry of the table. Top: Results with the baseline SIFT sampling density of 8 pixels and standard features. Bottom: Results with the set of parameters for SIFT sampling density and macrofeatures giving the best performance for sparse coding.

Discriminative Spatial Pyramid

- Tatsuya Harada, Yoshitaka Ushiku, Yuya Yamashita, and Yasuo Kuniyoshi. Discriminative Spatial Pyramid. In CVPR, 2011.

Purpose

Proposal of the discriminative, compact
and efficient Spatial Pyramid Representations

Background

Spatial Pyramid Representation (SPR)

- a widely used method for embedding spatial information
- good performance in terms of generic image recognition

Problems of SPR

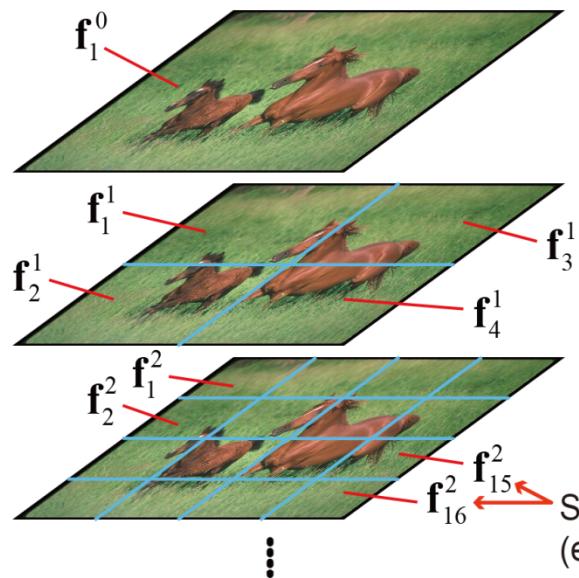
- a huge feature vector
- hand-crafted pyramid structure



Discriminative Spatial Pyramid

Weighted Spatial Pyramid Representation

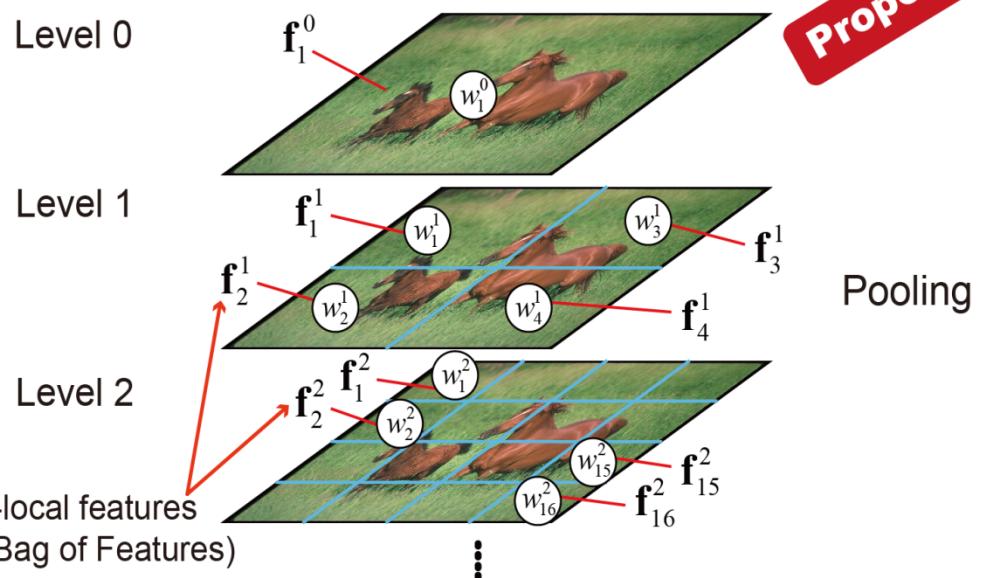
Spatial Pyramid Representation



Concatenation of semi-local features
(e.g. Bag of Features)

$$\mathbf{f}_S = \begin{pmatrix} \mathbf{f}_1^0 \\ \mathbf{f}_1^1 \\ \vdots \\ \mathbf{f}_{c(L-1)}^{L-1} \end{pmatrix} \rightarrow \text{Huge vector :-}$$

Weighted Spatial Pyramid Representation



Weighted sum of semi-local features

$$\mathbf{f}_W = w_1^0 \mathbf{f}_1^0 + w_2^0 \mathbf{f}_1^1 + \cdots + w_{c(L-1)}^{L-1} \mathbf{f}_{c(L-1)}^{L-1}$$

→ **Compact vector :-)**

But, how to obtain weights?

Discriminative Spatial Pyramid

