Semantic Indexing Using GMM Supervectors and Tree-structured GMMs

Nakamasa Inoue, Koichi Shinoda,
Department of Computer Science,
Tokyo Institute of Technology
Outline

- System overview
- Fast and high-performance semantic indexing system
  - 6 types of audio and visual features
  - Gaussian mixture model (GMM) supervectors
  - Tree-structured GMMs
- Best result: Mean InfAP = 17.3%
System Overview

- Fast and high-performance semantic indexing system

1) SIFT-Har
2) SIFT-Hes
3) SIFT-Dense
4) HOG-Dense
5) HOG-Sub
6) MFCC
System Overview

- Fast and high-performance semantic indexing system

1) SIFT-Har
2) SIFT-Hes
3) SIFTH-Dense
4) HOG-Dense
5) HOG-Sub
6) MFCC

Tree-structured GMMs

GMM supervectors

\[ \phi(X_s) \]

SVM score

Score fusion
Local Feature Extraction

1) SIFT-Har
   - Harris-affine detector: extension of Harris corner detector [Mikolajczyk, 2004]
   - Multi-frame (every other frame)

2) SIFT-Hes
   - Hessian-affine detector
   - Multi-frame (every other frame)

<table>
<thead>
<tr>
<th>Feature type</th>
<th>avg. #features per frame</th>
<th>avg. #features per shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT-Har</td>
<td>247</td>
<td>19,536</td>
</tr>
<tr>
<td>SIFT-Hes</td>
<td>240</td>
<td>18,986</td>
</tr>
</tbody>
</table>
Local Feature Extraction

3) SIFTH-Dense
   - SIFT + Hue histogram
   - 30,000 samples from a key-frame

4) HOG-Dense
   - 32 dimensional HOG
   - 10,000 samples from a key-frame

5) HOG-Sub
   - Dense HOG features extracted from temporal subtraction images
   - Capture movement
Local Feature Extraction

6) MFCC
- Mel-frequency cepstrum coefficients (MFCC)
- Audio features for speech recognition
- Targets: Speaking, Singing etc.

![Diagram of MFCC extraction process]

- Spectrum
- Filter bank
- MFCC(12)
- ΔMFCC(12)
- ΔΔMFCC(12)
- ΔLog-power(1)
- ΔΔLog-power(1)
System Overview

- Fast and high-performance semantic indexing system

1) SIFT-Har
2) SIFT-Hes
3) SIFTH-Dense
4) HOG-Dense
5) HOG-Sub
6) MFCC

Tree-structured GMMs

GMM supervectors

$\phi(X_s)$

Score fusion

SVM score

video (shot)
Gaussian Mixture Models (GMMs)

- Each shot is modeled by a GMM
  \[ X_F = \{ x_i \}_{i=1}^{n} \quad : \text{local features} \]
  \[ \theta = \{ w_k, \mu_k, \Sigma_k \}_{k=1}^{K} \quad : \text{GMM parameters} \]

- GMM parameters are estimated by using fast maximum a posteriori (MAP) adaptation

*Universal background model (UBM): a prior GMM which is estimated by using all video data.
Gaussian Mixture Models (GMMs)

- (Basic) MAP adaptation for mean vectors:

\[
\hat{\mu}_k = \frac{\tau \hat{\mu}_k^{(U)} + \sum_{i=1}^{n} c_{ik} x_i}{\tau + C_k}
\]

where

\[
c_{ik} = \frac{w_k N(x_i | \mu_k^{(U)}, \Sigma_k^{(U)})}{\sum_{k=1}^{K} w_k N(x_i | \mu_k^{(U)}, \Sigma_k^{(U)})}, \quad C_k = \sum_{i=1}^{n_s} c_{ik}
\]

*Universal background model (UBM): a prior GMM which is estimated by using all video data.

Computational cost: high
Gaussian Mixture Models (GMMs)

- $c_{ik}$: responsibility of component $k$ for $x_i$

\[ c_{ik} = \frac{w_k \mathcal{N}(x_i | \mu_k^{(U)}, \Sigma_k^{(U)})}{\sum_{k=1}^{K} w_k \mathcal{N}(x_i | \mu_k^{(U)}, \Sigma_k^{(U)})}, \]

- Tree-structured GMMs calculate $c_{ik}$ quickly!
Gaussian Mixture Models (GMMs)

- $c_{ik}$: responsibility of component $k$ for $x_i$

- Tree-structured GMMs calculate $c_{ik}$ quickly!
Tree-structured GMMs

- Calculate responsibilities $c_{ik}$ quickly.

$\begin{align*}
x_i & \quad \text{component } k \\
1 \text{st layer} & \\
2 \text{nd layer} & \\
3 \text{rd layer} & \\
\text{leaf layer} &
\end{align*}$

$c_{ik} : \text{responsibility of component } k \text{ for } x_i$

Tree-structured GMMs

- Leaf layer

Leaf node has a Gaussian of the UBM (prior GMM).

Tree-structured GMMs

- Non-leaf layers

Non-leaf node has a Gaussian that approximates its descendant Gaussians

Tree-structured GMMs

- Non-leaf layers

Non-leaf node has a Gaussian that approximates its descendant Gaussians

Tree-structured GMMs

- Non-leaf layers

Non-leaf node has a Gaussian that approximates its descendant Gaussians

Tree-structured GMMs

- Calculate responsibilities $c_{ik}$ quickly.

$\triangledown \text{Responsibility of component } k \text{ for } x_i$

1st layer
2nd layer
3rd layer
Leaf layer

Gaussian components

Fast MAP Adaptation

- Calculate responsibilities $c_{ik}$ quickly.

1. Initialize $V_A \leftarrow \{r\}$
   $V_A$ : a set of active nodes

Fast MAP Adaptation

- Calculate responsibilities $c_{ik}$ quickly.

2. Make children of $V_A$ active
3. Keep nodes if $c_i^{(v)} > c_{TH}$

Fast MAP Adaptation

- Calculate responsibilities $c_{ik}$ quickly.

2. Make children of $V_A$ active
3. Keep nodes if $c_i^{(v)} > c_{TH}$

Fast MAP Adaptation

- Calculate responsibilities $c_{ik}$ quickly.

2. Make children of $V_A$ active
3. Keep nodes if $c_{i(v)} > c_{TH}$

Fast MAP Adaptation

- Summary of the algorithm

\[ V_A : \text{a set of active nodes} \]
1. Initialize \( V_A \leftarrow \{r\} \)
   \( r \) : root node
2. Make children of \( V_A \) active
3. Calculate
   \[
   c_i^{(v)} = \frac{\tilde{w}^{(v)} g^{(v)}(x_i)}{\sum_{v \in V_A} \tilde{w}^{(v)} g^{(v)}(x_i)}
   \]
   and keep nodes active if \( c_i^{(v)} > c_{TH} \)
4. Go to 5 if all nodes in \( V_A \) are leaves, otherwise return to 2
5. Output GMM parameters

Fast MAP Adaptation

Summary of the algorithm

5. Output GMM parameters

\[
\hat{\mu}_k = \frac{\tau \hat{\mu}_k^{(u)} + \sum_{\hat{c}_{ik} \neq 0} \hat{c}_{ik} x_i}{\tau + \hat{C}_k}
\]

where

\[
\hat{c}_{ik} = \begin{cases} 
  c_i^{(\ell)} & (\ell \in V_A, \ g^{(\ell)} = g_k) \\
  0 & \text{(otherwise)}
\end{cases}
\]

\[
\hat{C}_k = \sum_{\hat{c}_{ik} \neq 0} \hat{c}_{ik}
\]

Fast MAP Adaptation

- Calculation time for MAP adaptation
  - 4.2 times faster than without tree-structured GMMs
  - No decrease in accuracy

Mean InfAP(%) on TRECVID 2010 dataset

<table>
<thead>
<tr>
<th>Feature</th>
<th>No tree</th>
<th>$T_{opt}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT-Har</td>
<td>6.30</td>
<td>6.32</td>
</tr>
<tr>
<td>SIFT-Hes</td>
<td>5.96</td>
<td>6.08</td>
</tr>
<tr>
<td>SIFTH-Dense</td>
<td>7.10</td>
<td>6.95</td>
</tr>
<tr>
<td>MFCC</td>
<td>1.99</td>
<td>2.00</td>
</tr>
<tr>
<td>Fusion</td>
<td>10.15</td>
<td>10.16</td>
</tr>
</tbody>
</table>

Optimized tree: the best tree in terms of calculation time on training data. Trees of depth at most 5 that have at most 5 children per node are tested.
GMM Supervector

- Combine normalized mean vectors.

\[ \phi(X_F) = \begin{pmatrix} \tilde{\mu}_1 \\ \tilde{\mu}_2 \\ \vdots \\ \tilde{\mu}_K \end{pmatrix} \]

where

\[ \tilde{\mu}_k = \frac{1}{\sqrt{\omega_k(\Sigma_k(\mu))}} \frac{1}{2} \hat{\mu}_k \]

(normalized mean)

\[ X_s = \{ x_i \}_{i=1}^{n_s} \]

UBM

Fast MAP adaptation

\( \phi(X_F) \)

GMM supervector
System Overview

- Fast and high-performance semantic indexing system

1) SIFT-Har
2) SIFT-Hes
3) SIFT-H-Dense
4) HOG-Dense
5) HOG-Sub
6) MFCC

Tree-structured GMMs

GMM supervectors $\phi(X_s)$

SVM score

Score fusion

SVM score
Score Fusion

- SVMs are trained with RBF-kernels

\[ k(X_F, X'_F) = \exp \left( -\gamma \| \phi(X_F) - \phi(X'_F) \|_2^2 \right) , \]

- Score fusion

Linear combination of SVM scores:

\[ f(X) = \sum_{F \in \mathcal{F}} \alpha_F f_F(X_F), \quad 0 \leq \alpha_F \leq 1, \quad \sum_F \alpha_F = 1 \]

where \( \mathcal{F} = \{ \text{SIFT-Har, SIFT-Hes, SIFTH-Dense, HOG-Dense, HOG-Sub, MFCC} \} \)

Combination coefficients \( \alpha_F \) are optimized on a validation set (IACC_1_tv10_training for training, and IACC_1_A for validation).
Experimental Condition

- **TokyoTech_Canon_1**
  6 features, 3 parameters for RBF-kernel
  (18 SVMs for one semantic concept)

\[
  f(X) = \sum_{h \in \{0.5, 1.0, 2.0\}} \alpha_{F}^{(h)} f_{F}^{(h)}(X_{F})
\]

\[
  \gamma = h^{-1} (h = 0.5, 1.0, 2.0)
\]

\[
  \mathcal{F} = \{\text{SIFT-Har, SIFT-Hes, SIF'TH-Dense, HOG-Dense, HOG-Sub, MFCC}\}
\]

- **TokyoTech_Canon_2**
  6 features, the parameter $h$ is fixed to 1.0
  (6 SVMs for one semantic concept)

\[
  f(X) = \sum_{F \in \mathcal{F}} \alpha_{F} f_{F}(X_{F})
\]

\[
  \mathcal{F} = \{\text{SIFT-Har, SIFT-Hes, SIF'TH-Dense, HOG-Dense, HOG-Sub, MFCC}\}
Experimental Condition

- **TokyoTech_Canon_3**
  Scores for all semantic concepts are combined:
  (i.e. 6 * 346 SVMs for one semantic concept)

  \[ g_s(X) = \sum_{S'} \beta_{S'} f_{S'}(X), \quad \sum_{S'} \beta_{S'} = 1 \]
  \[ f_{S'} : \text{score for concept } S' \]

- **TokyoTech_Canon_4**
  Additional training data from ImageNET
  (i.e. 6+1 SVMs for one semantic concept)

  \[ f(X) = \sum_{P \in \mathcal{F} \cup \{\text{HOG-Image}\}} \alpha_P f_P(X_P) \]

  \[ f_{\text{HOG-Image}} \] is trained on the TRECVID+ImageNET dataset with HOG-Dense features
## Results

<table>
<thead>
<tr>
<th>Run ID</th>
<th>Method</th>
<th>Mean InfAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TokyoTech_Canon_1</td>
<td>6 audio/visual GMM supervectors, 3 parameters for RBF-kernels</td>
<td>17.3%</td>
</tr>
<tr>
<td>TokyoTech_Canon_2</td>
<td>fixed parameter for RBF-kernels</td>
<td>17.3%</td>
</tr>
<tr>
<td>TokyoTech_Canon_3</td>
<td>2nd run + concept score fusion</td>
<td>16.4%</td>
</tr>
<tr>
<td>TokyoTech_Canon_4</td>
<td>2nd run + ImageNET images (Type D)</td>
<td>17.2%</td>
</tr>
</tbody>
</table>

![Graph showing TRECVID 2011 Semantic Indexing Runs](image.png)
InfAP by Semantic Concepts

![Graph showing InfAP by Semantic Concepts with various semantic concepts plotted against InfAP percentage. The graph includes markers for different TokyoTech and Canon datasets, as well as lines indicating Max and Median values.]
Which features are important?

- SIFT-Har: 12.44%
- +SIFT-H-Dense: 15.58%
- +MFCC: 16.59%
- +SIFT-Hes: 16.98%
- +HOG-Dense: 17.18%
- +HOG-Diff: 17.27%
Conclusion

- 6 types of audio and visual GMM supervectors
  Mean InfAP: 17.3%
  - Single feature: 12.4% (SIFT-Har (multi-frame))
  - 3 features: 16.6% (SIFT-Har, SIFTH-Dense, MFCC)
  - No audio: 16.4% (5 visual features)

- Fast MAP adaptation
  Tree-structured GMMs cut MAP adaptation costs.
  4.2 times faster than without tree-structured GMMs.

- Future work
  Human actions and event detection.
  Spatial and temporal localization.
Thank you!