PKU-IDM@TRECVID-CCD 2010: Copy Detection with Visual-Audio Feature Fusion and Sequential Pyramid Matching

General Coach: Wen Gao, Tiejun Huang
Executive Coach: Yonghong Tian, Yaowei Wang
Member: Yuanning Li, Luntian Mou, Chi Su, Menglin Jiang, Xiaoyu Fang, Mengren Qian

National Engineering Laboratory for Video Technology, Peking University
Outline

- Overview
  - Challenges
  - Our Results at TRECVID-CCD 2010

- Our Solution in the XSearch System
  - Multiple A-V Feature Extraction
  - Indexing with Inverted Table and LSH
  - Sequential Pyramid Matching
  - Automatic Verification and Fusion

- Analysis of Evaluation Results
- Demo
Challenges for TRECVID-CCD 2010

- **Dataset: Web video**
  - Poor quality
  - Diverse in content, style, frame rate, resolution...

- **Complex and severe transformations**
  - Audio: T5, T6 & T7
  - Video: T2, T6, T8 & T10

- Some non-copy queries are extremely similar with some ref. videos
Challenging Issues

- How to extract compact, “unique” descriptors (say, mediaprints) that are robust across a wide range of transformations?
  - Some mediaprints are robust against certain types but vulnerable to others; and vice versa.
  - Mediaprint ensembling: to enhance robustness and discriminability

- How to efficiently match mediaprints in a large-scale database?
  - Accurate and efficient mediaprint indexing
  - Trade off accuracy and speed

Four runs submitted

- “PKU-IDM.m.balanced.kraken”
- “PKU-IDM.m.nofa.kraken”
- “PKU-IDM.m.balanced.perseus”
- “PKU-IDM.m.nofa.perseus”

Excellent NDCR

- BALANCED profile, 39/56 top 1 “Actual NDCR”
- BALANCED profile, 51/56 top 1 “Optimal NDCR”
- NOFA profile, 52/56 top 1 “Actual NDCR”
- NOFA profile, 50/56 top 1 “Optimal NDCR”
Overview - Our Results at TRECVID-CCD (2)

- Comparable F1 score
  - Around 90%, with a few percent of deviation
  - No best, but most F1 scores are better than the medians

- Mean processing time is not satisfactory
  - Submission version: Worse than the median
  - Optimized version: Dramatically improved
Our System: XSearch

- Multiple complementary A-V features
- Inverted Table & LSH
- Sequential pyramid matching
- Verification and rank-based fusion
(1) Preprocessing

Audio
- Segmentation
  - 6s clips composed of 60ms frames, with 75% overlapping

Video
- Key-frame extraction
  - 3 frames/second
- Picture-In-Picture detection
  - Hough Transform
  - 3 frames: foreground, background and original frame
- Black frame detection
  - The percentage of pixels with luminance values equal to or smaller than a predefined threshold
- Flipping
  - Some key-frames are flipped to address mirroring in T8&T10
(2) Feature Extraction

- A single feature is typically robust against some transformations but vulnerable to others.

- Complementary features are extracted:
  - Audio feature (WASF)
  - Global visual feature (DCT)
  - Local visual feature (SIFT, SURF)

Contextual Local Features
- DVW, DVP, Bundled Feature

Regional Features
- Region-of-Interests, Segmentation, Multiple Instances

Global Features
- Color Histogram, Texture, Color Correlogram, edge-map

More Powerful Features
- Visual Sentence, Image Topic Model, etc.
- Refined
- Noisy
- Difficult
- Coarse
Audio Feature: WASF

Basic Idea

- An extension of MPEG-7 descriptor - Audio Spectrum Flatness (ASF) by introducing Human Audio System (HAS) functions to weight audio data
- Robust to sampling rate/amplitude/speed change/noise addition
- Extract from frequencies between 250 Hz and 3000 Hz
- 14-Dim WASF for a 60ms audio frame

Small-scale experiments show that WASF performs better than MFCC.

\[
WASF = \sqrt{\frac{1}{N} \sum_{i=0}^{n-1} w_i P_i}
\]

\[
w_i = \frac{P_i}{\sum_{k=0}^{n-1} P_k}
\]

N: the number of samples in each frequency band
P: the coefficient of power spectrum
Global Visual Feature: DCT

- **Basic Idea**
  - Robust to simple transformations (T4, T5, and T6)
  - Can handle complex transformations (T2, T3) after pre-processing
  - Low complexity (for all ref. data use 12 hours on 4-core PC)
  - Compact: 256 bits for a frame

![Flowchart of DCT process]

- Image → Convert to YUV → Obtain Y → Resize to 64x64 → Divide into 64 8x8 blocks
- Assign one bit to each block → Arrange into 4 rings → Compute 4 subband energy values → 2D-DCT

**DCT subband indexing**

\[
h(i,j) = \begin{cases} 
1 & B_i(S_j) \geq B_{i+1}(S_j) \quad (0 \leq i \leq 6, 0 \leq j \leq 3) \\
0 & B_i(S_j) < B_{i+1}(S_j) 
\end{cases}
\]

**DCT feature quantization**

\[
h(i,j) = \begin{cases} 
1 & B_i(S_j) \geq B_0(S_j) \quad (i = 63, 0 \leq j \leq 3) \\
0 & B_i(S_j) < B_0(S_j) 
\end{cases}
\]
Local Visual Feature: SIFT and SURF

Basic Idea

- Robust to T1 and T3, and to T2 after Picture-in-Picture detection
- Similar performance, but SIFT and SURF could be complementary
  - Copies that cannot be detected by SIFT could be detected by SURF, and vice versa
  - SURF descriptor is robust to flipping
- BoW employed over SIFT and SURF respectively
  - $K$-means for clustering local features into visual words ($k=400$)
- 64-Dim SURF and 128-Dim SIFT feature
Problems for SIFT and SURF

- Single BoW cannot preserve enough spatial information

Qi Tian, Build Contextual Visual Vocabulary for Large-Scale Image Applications, 2010.
Solution: Spatial Coding

- Use spatial, orientation and scale information
  - **Spatial quantization**: 0-20 for frame division of 1X1, 2X2, 4X4 cells
  - **Orientation quantization**: 0-17 for orientation division of 20° each
  - **Scale quantization**: 0-1 for small and big size

- To do in next step: Extract *local feature groups* for visual vocabulary generation to capture spatially contextual information[1]

(3) Indexing & Matching

- **Challenges**
  - **Accurate Search:** How to accurately locate the ref. items in a *similarity search* problem
  - **Scalability:** Quick matching in a very large ref. database
  - **Partial matching:** Whether a segment of the query item matches a segment of one or more ref. items in the database

- **Our Solutions**
  - Inverted table for accurate search
  - Local sensitive hashing for approximate search
  - Sequential Pyramid Matching (SPM) for coarse-to-fine search
Inverted Table: for Accurate Search

- Key-frame retrieval using inverted index
Local Sensitive Hashing: for Approximate Search

Basic Idea

- If two points are close together, they will remain so after a "projection" operation.
- To hash a large reference database into a much-smaller-size bucket of match candidates, then use a linear, exhaustive search to find the points in the bucket that are closest to the query point.

Used on WASF and DCT

SPM: for Coarse-to-Fine Search

- Keyframe-based solution: from frame matching to segment matching
- SPM: To **filter out** the mismatched candidates by frame-level voting and **align** the query video with the reference video
- **Steps**
  1. Frame matching: Find top k ref. frames for each query frame
  2. Subsequence location: Identify the first and the last matched key-frames of a candidate reference video and a query video
  3. Alignment: Slide the subsequence of the query over the subsequence of the candidate reference to align two sequences
  4. Multi-granularity fusion: Evaluate the similarity using different weights for different granularities
SPM : for Coarse-to-Fine Search

Query sequence:

Level 1: MatchingPairs × 1

Level 2: + MatchingPairs × 1/2

Level 3: + MatchingPairs × 1/4
(4) Verification and Fusion

- An additional **Verification** module
  - BoW representation can cause an increase in false alarm rate
  - Matches of SIFT and SURF points (instead of BoW) are used to verify result items that are only reported by a single basic detector
  - The verification method: perform point matching and check the spatial consistency
  - The final similarity is calculated by counting the matching points.
  - Only used for the “perseus” submissions

- An example

  ![Example Image](image-url)

  TP when matching with BoW

  FA after verification
(4) Verification and Fusion

- **Rank-based** fusion for final detection results (ad hoc!)
  - Intersection of detection results by any two basic detectors are assumed to be copies with very high probability
  - Rule-based post-processing is adopted to filter out those results below a certain threshold
Analysis of Evaluation Results

- **NDCR**
  - BALANCED Profile: Actual NDCR
  - BALANCED Profile: Optimal NDCR
  - NOFA Profile: Actual NDCR
  - NOFA Profile: Optimal NDCR

- **F1**

- **Processing Time**
  - Submission version
  - Optimized version
BALANCED Profile: Actual NDCR

39/56 top 1 “Actual NDCR”
- Perseus: 31
- Kraken: 12 (4 overlapped)

Using log-value
BALANCED Profile: Optimal NDCR

- 51/56 top 1 “Optimal NDCR”
  - Perseus: 47
  - Kraken: 16 (12 overlapped)

Using log-value
NOFA Profile: Actual NDCR

☐ 52/56 top 1 “Actual NDCR”
  ☐ Perseus: 52
  ☐ Kraken: 4 (4 overlapped)
NOFA Profile: Optimal NDCR

- 50/56 top 1 “Optimal NDCR”
  - Perseus: 50
  - Kraken: 4 (4 overlapped)
Lesson Learned

- Multiple complementary A-V features
  - Feature refinement is very important
- SPM to guarantee a high recall
- Verification to ensure precision
  - SIFT and SURF matches (instead of BoWs) are used to filter candidates with both similarities of SIFT and SURF smaller than a threshold
- Rank-based fusion to further sift FAs

- However, at the cost of F1 and mean processing time
F1 for both Profiles

- Comparable mean F1 score
  - Around 90%, with a few percent of deviation
- A large room for improvement compared to the best
  - Keyframe-based solution introduces sampling
  - For complicated transformations, the similarity is low. As a result, very high TP causes a drop at F1.
  - SPM based on top k similar frames; frames not in top k not examined.
Mean Processing Time

- **Submission version:** Worse than the median
  - Time-consuming of multi-features: esp. local visual features extraction
  - Not-optimal Programming: Single-processing, single-threading
  - Low-performance Machines: <=8 cores PC Servers with <=8G M

- **Optimized version:** Dramatically improved
  - Optimization of local features (SIFT & SURF)
  - Multi-threading, Multi-processing
  - High-perf Server (32 cores, 32G M)
How to further improve the efficiency?

- **Compact and robust descriptors**
  - Compressed Histogram of Gradient (CHoG): approximate 50 bits
  - Compressed SIFT descriptor: 2 bits/dimension (128 in total)

- **Configurable sets of features**
  - According to different datasets or transformations, the system adopts different sets of features

- **Fast, accurate indexing and matching**
  - Pre-computed and cached similarity in inverted table

- **CCD: Computing-Intensive Application**
  - A Possible Solution: Multimedia Service Cloud?
Demo
THANKS

Member: Yonghong Tian, Yaowei Wang
        Yuanning Li, Luntian Mou, Chi Su,
        Menglin Jiang, Xiaoyu Fang, Mengren Qian

National Engineering Laboratory for Video Technology, Peking University