Robust Video Fingerprinting

Mu Li

iPAL Group Meeting

February 17, 2012
Video hashing: Randomized video dimensionality reduction

\[ \mathcal{V} \rightarrow h_K(\mathcal{V}) \]

Anti-piracy video search: Youtube.com

Requirements on video hashing vector:

- **Robustness:** For a given \( \theta > 0 \) and \( \forall \mathcal{V}, A(\mathcal{V}) \):

  \[
  Pr\{|\| h_K(\mathcal{V}) - h_K(A(\mathcal{V})) \| < \tau \} \geq 1 - \theta
  \]

- **Security:** Prevent adversarial attack \( \rightarrow h_K(\mathcal{V}) \) with varying \( K \in \mathcal{K} \) should exhibit high entropy over \( \mathcal{K} \)

- **Computational efficiency:** Realtime computation.
Introduction

- Video hashing: Randomized video dimensionality reduction
  \[ V \rightarrow h_K(V) \]

- Anti-piracy video search: Youtube.com

- Requirements on video hashing vector:
  - **Robustness**: For a given \( \theta > 0 \) and \( \forall V, A(V) \):
    \[ Pr\{\| h_K(V) - h_K(A(V)) \| < \tau \} \geq 1 - \theta \]
  - **Security**: Prevent adversarial attack \( \rightarrow h_K(V) \) with varying \( K \in \mathcal{K} \) should exhibit high entropy over \( \mathcal{K} \)
  - **Computational efficiency**: Realtime computation.
Video hashing: State of the art

- **Two types of methods:**
  1. Frame based hashing: Extension of image hashing, including CGO, RASH, etc.
  2. Spatial-temporal hashing: Both spatial and temporal information, including 3-D DCT.

- **Main challenges:**
  1. Frame based methods suffer from temporal video attacks, such as frame rate change, random frame substitution, etc.
  2. 3-D DCT suffers under geometric video attacks, like frame rotation, cropping.
Video hashing: State of the art

- Two types of methods:
  1. Frame based hashing: Extension of image hashing, including CGO, RASH, etc.
  2. Spatial-temporal hashing: Both spatial and temporal information, including 3-D DCT.

- Main challenges:
  1. Frame based methods suffer from temporal video attacks, such as frame rate change, random frame substitution, etc.
  2. 3-D DCT suffers under geometric video attacks, like frame rotation, cropping.
Typical algorithms: CGO\(^1\), RASH\(^2\) and 3-D DCT\(^3\)

- **Centroid of Gradient Orientations (CGO):** Extract direction information from difference of adjacent pixel values.
- **Radial projection based Hash (RASH):** Compute the variance of pixel values on a set of lines articulated around the center of the frame.
- **3-D DCT:** Select low frequency 3-D DCT coefficients as hashing values.

---

\(^1\) Lee et al., IEEE Trans. Circuits Syst. Video Technol., 2008
\(^3\) Coskun et al., IEEE Trans. Multimedia, 2006

02/17/2012
iPAL Group Meeting
Motivation for our algorithm

- Frame based methods don’t capture temporal evolution information
- 3-D DCT → spatial-temporal, but coefficients lack interpretation
  - Proposed idea: Use tensor factorization to compute video hashing:
    - An $N$-th order tensor is a $N$-dimensional array $\mathbf{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$
    - First-order tensor → vector, second-order tensor → matrix.
    - Video → modeled well by a third-order tensor.
Motivation for our algorithm

- Frame based methods don’t capture **temporal evolution information**
- 3-D DCT → spatial-temporal, but coefficients lack interpretation
- **Proposed idea:** Use tensor factorization to compute video hashing:
  - An $N$-th order tensor is a $N$-dimensional array $\mathbf{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$
  - First-order tensor → vector, second-order tensor → matrix.
  - Video → modeled well by a third-order tensor.
Robust video hashing via randomized tensor approximations\textsuperscript{4,5,6}

\textsuperscript{4} Supported by a gift from Youtube.
\textsuperscript{5} M. Li and V. Monga, IEEE MMSP, Oct. 2011 - Top 10\% paper award
Applying tensor factorization to video hashing

- Parafac tensor factorization: an extension of matrix SVD and PCA.

\[ y \approx a_1 c_1 b_1 + a_2 c_2 b_2 + \ldots + a_r c_r b_r \]

- Video size: 128 × 128 × 128
- Rank-30 tensor factorization

02/17/2012 iPAL Group Meeting
Validating hash robustness

- Attack: Temporal sub-sampling by a factor of 4.

(a) LRTA

(b) 3-D DCT

(c) CGO

(d) RASH
Tensor factorization and randomization

- Different types of tensor factorizations
- **PARAFAC** factorization used in our algorithm
- Rank-$r$ factorization of tensor $\mathbf{Y} \in \mathbb{R}^{I \times J \times K}$:

  \[
  \mathbf{Y} \approx [\lambda; \mathbf{A}, \mathbf{B}, \mathbf{C}] \triangleq \sum_{i=1}^{r} \lambda_i \mathbf{a}_i \odot \mathbf{b}_i \odot \mathbf{c}_i
  \]

- Security of hashing vector: randomly select overlapping sub-cubes.
Algorithm formulation and performance evaluation

**Algorithm 1 Tensor factorization-based video hashing**

Given: Query video.

1. **Normalized video:** Temporal subsampling, transform to grayscale, normalize each frame to size $I \times J \rightarrow$ normalized video $I \times J \times K$.
2. **Random sub-cubes:** Use secret key to randomize locations of overlapping sub-cubes of size $M \times N \times P$ to cover entire video.
3. For each sub-cube, calculate its rank-1 parafac tensor factorization, getting vectors $x_i \in \mathbb{R}^M$, $y_i \in \mathbb{R}^N$, $z_i \in \mathbb{R}^P$, $i = 1 \ldots Q$.
4. **Final hash:** $h = \left[ \frac{\sum_{i=1}^Q x_i}{Q}; \frac{\sum_{i=1}^Q y_i}{Q}; \frac{\sum_{i=1}^Q z_i}{Q} \right] \in \mathbb{R}^{M+N+P}$.

- Example: each sub-cube is $42 \times 42 \times 42 \Rightarrow$ final hash $h \in \mathbb{R}^{126}$.
- ROC often used to evaluate hashing algorithms:

  $$P_M(\tau) = Pr(\| h_K(V) - h_K(A(V)) \| > \tau)$$
  $$P_{FA}(\tau) = Pr(\| h_K(V) - h_K(A(V')) \| < \tau).$$
Algorithm formulation and performance evaluation

Algorithm 2 Tensor factorization-based video hashing

Given: Query video.

1. **Normalized video**: Temporal subsampling, transform to grayscale, normalize each frame to size $I \times J \to$ normalized video $I \times J \times K$.

2. **Random sub-cubes**: Use secret key to randomize locations of overlapping sub-cubes of size $M \times N \times P$ to cover entire video.

3. For each sub-cube, calculate its rank-1 parafac tensor factorization, getting vectors $x_i \in \mathbb{R}^M, y_i \in \mathbb{R}^N, z_i \in \mathbb{R}^P, i = 1 \ldots Q$.

4. **Final hash**: $h = \left[ \frac{\sum_{i=1}^{Q} x_i}{Q}, \frac{\sum_{i=1}^{Q} y_i}{Q}, \frac{\sum_{i=1}^{Q} z_i}{Q} \right] \in \mathbb{R}^{M+N+P}$.

- Example: each sub-cube is $42 \times 42 \times 42 \Rightarrow$ final hash $h \in \mathbb{R}^{126}$.
- ROC often used to evaluate hashing algorithms:

$$P_M(\tau) = Pr(\| h_K(V) - h_K(A(V)) \| > \tau)$$

$$P_{FA}(\tau) = Pr(\| h_K(V) - h_K(A(V')) \| < \tau).$$
Performance comparisons under common attacks

- Measure $D$:

$$D = \frac{\sum_{i=1}^{n} \|h_K(V_i) - h_K(A(V_i))\|}{\frac{n}{2} \sum_{i=1}^{n} \sum_{j=1}^{i-1} \|h_K(V_i) - h_K(A(V_j))\|}$$

Table: Normalized hash deviation under different attacks

<table>
<thead>
<tr>
<th>Attack</th>
<th>LRTA</th>
<th>DCT</th>
<th>CGO</th>
<th>RASH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compression</td>
<td>0.0097</td>
<td>0.0126</td>
<td>0.1281</td>
<td>0.0091</td>
</tr>
<tr>
<td>Contrast enhancement</td>
<td>0.4208</td>
<td>0.4693</td>
<td>0.4766</td>
<td>1.1265</td>
</tr>
<tr>
<td>Blurring</td>
<td>0.1293</td>
<td>0.0289</td>
<td>0.4137</td>
<td>0.3096</td>
</tr>
<tr>
<td>AWGN</td>
<td>0.3487</td>
<td>0.2138</td>
<td>0.5247</td>
<td>0.6023</td>
</tr>
<tr>
<td>Frame rotation</td>
<td>0.2003</td>
<td>0.4031</td>
<td>0.8924</td>
<td>0.1048</td>
</tr>
<tr>
<td>Frame cropping</td>
<td>0.6114</td>
<td>0.9103</td>
<td>0.6591</td>
<td>0.3526</td>
</tr>
<tr>
<td>Frame rate change (Factor 2)</td>
<td>0.1043</td>
<td>0.0810</td>
<td>0.2463</td>
<td>0.1770</td>
</tr>
<tr>
<td>Frame rate change (Factor 4)</td>
<td>0.1773</td>
<td>0.1857</td>
<td>0.3429</td>
<td>0.2855</td>
</tr>
<tr>
<td>Frame rate change (Factor 8)</td>
<td>0.2721</td>
<td>0.4089</td>
<td>0.4503</td>
<td>0.5000</td>
</tr>
<tr>
<td>Random frame deletion</td>
<td>0.1301</td>
<td>0.1719</td>
<td>0.2635</td>
<td>0.1600</td>
</tr>
</tbody>
</table>
Performance comparisons under random local bending

- Rash: most robust to bending, tensor algorithm: second best.
Performance comparisons under random frame dropping

Frame based hashing methods (RASH, CGO) suffer from temporal attack, tensor method is the best.
Performance comparisons under a composite attack

- Tensor method is the best; CGO and RASH → worse than random guess.
Detection-theoretic analysis

- Formulate video hashing as a binary hypothesis testing problem:
  - $y \rightarrow$ hash of query video
  - $x \rightarrow$ hash of reference video
  - $z \rightarrow$ hash of a "perceptually" distinct video
  - $n \rightarrow$ noise
  - $w \triangleq y - x \in \mathbb{R}^L$.

\[ H_0 : w = z - x + n \sim f_{w|x,H_0}(w) \]
\[ H_1 : w = n \sim f_{w|x,H_1}(w) = f_{w|H_1}(w). \]
Detection-theoretic analysis

- Hashing vectors of our algorithm can be modeled well by a Gaussian Mixture Model:

$$f_{w|H_1}(w) = \sum_{i=1}^{M} c_i r_i(w); \quad \sum_{i=1}^{M} c_i = 1, \quad c_i \in \mathbb{R}$$

$$f_{w|H_0}(w) = \sum_{i=1}^{N} d_i q_i(w); \quad \sum_{i=1}^{N} d_i = 1, \quad d_i \in \mathbb{R}$$

$$r_i(w) \sim \mathcal{N}(\mu_{1i}, \Sigma_{1i}), i = 1 \ldots M, \quad \mu_{1i} \in \mathbb{R}^L, \Sigma_{1i} \in \mathbb{R}^{L \times L}$$

$$q_i(w) \sim \mathcal{N}(\mu_{0i}, \Sigma_{0i}), i = 1 \ldots N, \quad \mu_{0i} \in \mathbb{R}^L, \Sigma_{0i} \in \mathbb{R}^{L \times L}.$$

02/17/2012  iPAL Group Meeting
Expressions for error probabilities

False alarm probability can be shown to be:

\[ P_{FA}(\tau) \approx \sum_{i=1}^{N} d_i \gamma \left( \frac{1}{2} \left( \sum_{j=1}^{L} \lambda_{0ij}^2 + 2 \sum_{j=1}^{L} \lambda_{0ij}^2 \varphi_j \right)^3 - \frac{1}{2} \left( \sum_{j=1}^{L} \lambda_{0ij}^3 + 3 \sum_{j=1}^{L} \lambda_{0ij}^3 \varphi_j \right)^2 \right) \]

where \( \varphi_j = \xi_j^2 \), \( \xi = P_{0i}(E^T)^{-1} \mu_{0i} \), \( P_{0i}^T \in \mathbb{R}^{L \times L} \) is the normalized eigenvector matrix of \( \Sigma_{0i} \), \( E^T \in \mathbb{R}^{L \times L} \) is the Cholesky decomposition of \( \Sigma_{0i} \) and \( \lambda_{0i1}, \ldots, \lambda_{0iL} \) are eigenvalues of \( \Sigma_{0i} \). Similarly,

\[ P_M(\tau) \approx \sum_{i=1}^{M} c_i - \sum_{i=1}^{M} c_i \gamma \left( \frac{1}{2} \left( \sum_{j=1}^{L} \lambda_{1ij}^2 + 2 \sum_{j=1}^{L} \lambda_{1ij}^2 \delta_j \right)^3 - \frac{1}{2} \left( \sum_{j=1}^{L} \lambda_{1ij}^3 + 3 \sum_{j=1}^{L} \lambda_{1ij}^3 \delta_j \right)^2 \right) \]

\[ + \frac{1}{2} \left( \tau^2 - \sum_{j=1}^{L} \lambda_{1ij} - \sum_{j=1}^{L} \lambda_{1ij} \delta_j \right) \left( \sum_{j=1}^{L} \lambda_{1ij}^2 + 2 \sum_{j=1}^{L} \lambda_{1ij}^2 \delta_j \right) \]

\[ \gamma \left( \frac{1}{2} \left( \sum_{j=1}^{L} \lambda_{1ij}^3 + 3 \sum_{j=1}^{L} \lambda_{1ij}^3 \delta_j \right)^2 \right) \]
Expressions for error probabilities

False alarm probability can be shown to be:

\[
P_{FA}(\tau) \approx \sum_{i=1}^{N} d_i \gamma \left( \frac{1}{2} \left( \sum_{j=1}^{L} \lambda_{1ij}^2 + 2 \sum_{j=1}^{L} \lambda_{1ij}^2 \delta_j \right)^2, \frac{1}{2} \left( \sum_{j=1}^{L} \lambda_{1ij}^2 + 2 \sum_{j=1}^{L} \lambda_{1ij}^2 \varphi_j \right)^2 \right) \left( \sum_{j=1}^{L} \lambda_{2ij}^2 + 2 \sum_{j=1}^{L} \lambda_{2ij}^2 \delta_j \right)^3 \left( \sum_{j=1}^{L} \lambda_{3ij}^2 + 3 \sum_{j=1}^{L} \lambda_{3ij}^2 \delta_j \right)^2 + \frac{1}{2} \left( \tau^2 - \sum_{j=1}^{L} \lambda_{0ij} - \sum_{j=1}^{L} \lambda_{0ij} \varphi_j \right) \left( \sum_{j=1}^{L} \lambda_{2ij}^2 + 2 \sum_{j=1}^{L} \lambda_{2ij}^2 \delta_j \right) \right) \right),
\]

where \( \varphi_j = \xi_{2j}^2, \xi = P_{0i} (E^T)^{-1} \mu_{0i}, P_{0i} \in \mathbb{R}^{L \times L} \) is the normalized eigenvector matrix of \( \Sigma_{0i} \), \( E^T \in \mathbb{R}^{L \times L} \) is the Cholesky decomposition of \( \Sigma_{0i} \) and \( \lambda_{0i1}, \ldots, \lambda_{0iL} \) are eigenvalues of \( \Sigma_{0i} \). Similarly,

\[
P_{M}(\tau) \approx \sum_{i=1}^{M} c_i - \sum_{i=1}^{M} c_i \gamma \left( \frac{1}{2} \left( \sum_{j=1}^{L} \lambda_{1ij}^2 + 2 \sum_{j=1}^{L} \lambda_{1ij}^2 \delta_j \right)^2, \frac{1}{2} \left( \sum_{j=1}^{L} \lambda_{1ij}^2 + 2 \sum_{j=1}^{L} \lambda_{1ij}^2 \delta_j \right)^2 \right) \left( \sum_{j=1}^{L} \lambda_{2ij}^2 + 2 \sum_{j=1}^{L} \lambda_{2ij}^2 \delta_j \right)^3 \left( \sum_{j=1}^{L} \lambda_{3ij}^2 + 3 \sum_{j=1}^{L} \lambda_{3ij}^2 \delta_j \right)^2 + \frac{1}{2} \left( \tau^2 - \sum_{j=1}^{L} \lambda_{1ij} - \sum_{j=1}^{L} \lambda_{1ij} \delta_j \right) \left( \sum_{j=1}^{L} \lambda_{2ij}^2 + 2 \sum_{j=1}^{L} \lambda_{2ij}^2 \delta_j \right) \right) \right),
\]
Accuracy of theoretical approximations

(a) Compression and AWGN.

(b) Random local bending.

(c) Random frame dropping from both halves.

(d) Rotation by 5 deg and temporal sampling with factor 4.
Conclusions

- Robust video hashing algorithm based on PARAFAC low-rank tensor approximations
- Effectively captures both spatial as well as temporal components of video content
  - Enables higher level of robustness while maintaining a given level of discriminability
- Analytical expressions for detection-theoretic error probabilities match up closely against empirically obtained ROCs.