REVERSE ENGINEERING OF VIDEO CONTENT FOR FORENSIC ANALYSIS

Paolo Bestagini
Ph.D.

paolo.bestagini@polimi.it
Motivations
Motivations

Diffusion of multimedia sharing platforms
Motivations

• Diffusion of multimedia sharing platforms
Motivations

- Diffusion of multimedia **sharing platforms**
- Huge amount of user-generated content
Motivations

- Diffusion of multimedia sharing platforms
- Huge amount of user-generated content
- Availability of user-friendly editing software
Motivations

• Diffusion of multimedia **sharing platforms**
  • Huge amount of user-generated content

• Availability of user-friendly **video-editing software**
Motivations

- Diffusion of multimedia **sharing platforms**
  - Huge amount of user-generated content
- Availability of user-friendly **video-editing software**
  - Easy to tamper with videos
Motivations

- Diffusion of multimedia sharing platforms
- Huge amount of user-generated content
- Availability of user-friendly video-editing software

We cannot trust what we see!
Motivations

- Diffusion of multimedia sharing platforms
- Huge amount of user-generated content
- Availability of user-friendly video-editing software
- Easy to tamper with videos

We cannot trust what we see!
Motivations

- Diffusion of multimedia sharing platforms
- Huge amount of user-generated content
- Availability of user-friendly video-editing software
- Easy to tamper with videos

We cannot trust what we see!
Motivations

• Diffusion of multimedia sharing platforms
• Huge amount of user-generated content
• Availability of user-friendly video-editing software

We cannot trust what we see!
Motivations

• Diffusion of multimedia sharing platforms
• Huge amount of user-generated content
• Availability of user-friendly video-editing software

We cannot trust what we see!
Motivations

- Diffusion of multimedia sharing platforms
- Huge amount of user-generated content
- Availability of user-friendly video-editing software

We cannot trust what we see!
Motivations

• Diffusion of multimedia **sharing platforms**
  • Huge amount of user-generated content

• Availability of user-friendly **video-editing software**
  • Easy to tamper with videos

• We **cannot trust** what we see!
Motivations

- Diffusion of multimedia sharing platforms
  - Huge amount of user-generated content
- Availability of user-friendly video-editing software
- Easy to tamper with videos
- We cannot trust what we see!
Objectives
Objectives

• **What can we do?**
  • To develop a series of **blind** algorithms and tools for **video forensic analyses** working in a real world scenario.
Objectives

• What can we do?
  • To develop a series of **blind** algorithms and tools for **video forensic analyses** working in a real world scenario.

• How?
  • Every **non-reversible operation** leaves peculiar footprints.
  • **Footprints** as an asset.
Objectives

• What can we do?
  • To develop a series of blind algorithms and tools for video forensic analyses working in a real world scenario.

• How?
  • Every non-reversible operation leaves peculiar footprints.
  • Footprints as an asset.

Fig. 4: Original (a) and compressed (b) frames of a standard video sequence. The high compression rate is responsible for blocking artifacts. Generated the video content; iii) estimate the quality of the reconstructed video without the availability of the original source. In the literature, the methods aiming to estimate different coding parameters and syntax elements characterizing the adopted codec can be grouped into three main categories, which are further described below: i) approaches detecting block boundaries; ii) approaches estimating the quantization parameters, and iii) approaches estimating the motion vectors.

1) Block detection:
Most video coding architectures encode frames on a block-by-block basis. For this reason, artifacts at block boundaries can be exploited to reveal traces of previous compression steps. Typical blocking artifacts are shown in Fig 4. Identifying block boundaries allows also estimating the block size. It is possible to detect block-wise coding operations by checking local pixel consistency, as shown in [24], [25]. There, the authors evaluate whether the statistics of pixel differences across blocks differ from those of pixels within the same block. In this case, the image is supposed to be the result of block-wise compression. In [48], the block size in a compressed video sequence is estimated by analyzing the reconstructed picture in the frequency domain and detecting those peaks that are related to discontinuities at block boundaries, rather than intrinsic features of the underlying image. However, some modern video coding architectures (including, e.g., H.264/AVC as well as the recent HEVC standard under development) enable to use a deblocking filter to smooth artifacts at block boundaries, in addition to variable block sizes (also with non-square blocks). In these situations, traditional block detection methods fail, leaving this as an open issue for further investigations.

2) Quantization step detection:
Scalar quantization in the transform domain leaves a very common footprint in the histogram of transform coefficients. Indeed, the histogram of each coefficient \( Y_r(i, j) \) shows a typical comb-like distribution, in which the peaks are spaced apart by \( \Delta(i, j) \), instead of a
Objectives

• What can we do?
  • To develop a series of **blind** algorithms and tools for **video forensic analyses** working in a real world scenario.

• How?
  • Every **non-reversible operation** leaves peculiar footprints.
  • **Footprints** as an asset.
Objectives

• **What can we do?**
  • To develop a series of **blind** algorithms and tools for **video forensic analyses** working in a real world scenario.

• **How?**
  • Every non-reversible operation leaves peculiar footprints.
  • Footprints as an asset.

![Original and compressed frames of a standard video sequence.](image)

- Original (a) and compressed (b) frames of a standard video sequence. The high compression rate is responsible for blocking artifacts.
Objectives

• What can we do?
  • To develop a series of **blind** algorithms and tools for **video forensic analyses** working in a real world scenario.

• How?
  • Every **non-reversible operation** leaves peculiar footprints.
  • **Footprints** as an asset.
Objectives

• **What can we do?**
  • To develop a series of forensic analyses tools and approaches working in a real-world scenario.

• **How?**
  • Every non-reversible operation leaves peculiar footprints.
  • **Footprints** as an asset.
Objectives

• What can we do?
  • To develop a series of **blind** algorithms and tools for **video forensic analyses** working in a real world scenario.

• How?
  • Every **non-reversible operation** leaves peculiar footprints.
  • **Footprints** as an asset.
Analysis on a single video
Analysis on a single video
Analysis on a single video

- Coding-based footprints
- Editing-based footprints
- Acquisition-based footprints

Video forensics
Analysis on a single video

- Coding-based footprints
- Editing-based footprints
- Acquisition-based footprints

Video forensics
Analysis on a single video

- Coding-based footprints
- Editing-based footprints
- Acquisition-based footprints

Video forensics
Analysis on a single video

- Coding-based footprints
- Editing-based footprints
- Acquisition-based footprints

Video forensics
Analysis on a single video

- **Number** of compressions
- **Type of codec**
- **Coding-based footprints**
- **Editing-based footprints**
- **Acquisition-based footprints**

**Video forensics**
Analysis on a single video

- **Coding-based footprints**
  - Number of compressions
  - Type of codec

- **Image splicing**
- **Video copy-paste**
- **Video copy-move**
- **Editing-based footprints**
- **Acquisition-based footprints**

- **Video forensics**
Analysis on a single video

- **Coding-based footprints**
  - Number of compressions
  - Type of codec

- **Editing-based footprints**
  - Image splicing
  - Video copy-paste

- **Acquisition-based footprints**
  - Video copy-move
  - Re-capture
  - Acquisition-based footprints

- **Video forensics**
Coding-based footprints

- Number of compressions
- Type of codec
Coding-based footprints

- Videos are often **encoded multiple times** during their life-time
  - Information about acquisition device
  - The number of compression steps is an indicator of a video reliability
How many times has the video been compressed?

[Bestagini et al. MMSP 2012]
Coding-based footprints: multiple compression

- **Video coding:**
  - **Temporal** redundancy

- **Spatial** redundancy
Coding-based footprints: multiple compression

- Benford’s law:

- The distribution of the first digit (FD) of a single quantized DCT coefficient approximatively follows Benford’s law:

\[ p(m) = K \log_{10} \left( 1 + \frac{1}{\alpha + m^\beta} \right), \text{ with } m = 1, \ldots, 9. \]

- When multiple quantized, this law does not hold!
Coding-based footprints: multiple compression

- Single quantized:
  \[ \Delta = 0.1 \]

- Double quantized:
  \[ \Delta_1 = 0.1, \Delta_2 = 0.8 \]
Coding-based footprints: multiple compression

• **Method 1:**
  - Compute FD histograms for a set of DCT frequencies.
  - Train a set of SVMs.
  - Combine SVMs outputs.

• **Results:**
  - Up to three compressions successfully detected

![ROC curves comparing](image)

**Table 3.2.** Confusion matrix for coding parameters $N$, $N^*$

<table>
<thead>
<tr>
<th>$N$, $N^*$</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>2</td>
<td>0.00%</td>
<td>73.89%</td>
<td>26.11%</td>
</tr>
<tr>
<td>3</td>
<td>0.00%</td>
<td>22.22%</td>
<td>77.78%</td>
</tr>
</tbody>
</table>
Which codec has been used to encode a double-compressed video?

[Bestagini et al. ICASSP 2012]
[Bestagini et al. EUVIP 2013]
[Bestagini et al. TIP 2016]
Coding-based footprints: codec identification

- **Idempotency property:**
  - if we re-quantize an already quantized signal with the same quantization step, the signal does not change

\[
\hat{X}_1 = Q_{\Delta_1}(X) = \Delta_1 \left\lfloor \frac{X}{\Delta_1} \right\rfloor \\
\hat{X}_2 = Q_{\Delta_2}(\hat{X}_1) = \Delta_1 \left\lfloor \frac{\Delta_1 \left\lfloor \frac{X}{\Delta_1} \right\rfloor}{\Delta_1} \right\rfloor = \hat{X}_1
\]

- This is partly true also for video codecs

![Diagram of coding process](image)
Coding-based footprints: codec identification

• Main idea:

\[ X \xrightarrow{\text{Codec 1}} \hat{X}_1 \xrightarrow{\text{Codec 1}} \hat{X}_2 \cong \hat{X}_1 \]

\[ X \xrightarrow{\text{Codec 1}} \hat{X}_1 \xrightarrow{\text{SMALL NOISE}} \hat{X}_2 \cong \hat{X}_1 \xrightarrow{\text{Codec 1}} \hat{X}_3 \cong \hat{X}_1 \cong \hat{X}_2 \]
Coding-based footprints: codec identification

• Main idea:
Coding-based footprints: codec identification

• Main idea:

\[ \hat{X}_2 \approx \hat{X}_1 \]

\[ \hat{X}_3 \approx \hat{X}_1 \approx \hat{X}_2 \]

• Approach:

\[ \hat{X}_3 \approx \hat{X}_2 \]

controlled parameters
Coding-based footprints: codec identification

Parameters:
GOP only Intra
QP \( QP_{\text{min}} : QP_{\text{max}} \)

\[
\text{PSNR}(\hat{X}_2, \hat{X}_3)
\]

\[
\hat{X}_3 \rightarrow \text{Frame number} \rightarrow \text{QP}
\]

\[
\hat{X}_2 \rightarrow \text{Frame number} \rightarrow \text{QP}
\]

\[
\hat{X}_1 \rightarrow \text{Frame number} \rightarrow \text{QP}
\]

\[
X \rightarrow \text{MPEG2} \rightarrow \hat{X}_1 \rightarrow \text{AVC} \rightarrow \hat{X}_2 \rightarrow \text{MPEG2} \rightarrow \hat{X}_3
\]

\[
\hat{X}_2 \rightarrow \text{MPEG4} \rightarrow \hat{X}_3
\]

\[
\hat{X}_1 \rightarrow \text{AVC}
\]
Coding-based footprints: codec identification

Correct

Incorrect

![Correct](image1)

![Incorrect](image2)

![Graph](image3)
Coding-based footprints: codec identification

Correct

Incorrect

Local Maxima
Coding-based footprints: codec identification

- Results:

<table>
<thead>
<tr>
<th>c_1</th>
<th>MPEG-2</th>
<th>MPEG-4</th>
<th>AVC</th>
<th>DIRAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPEG-2</td>
<td>0.94</td>
<td>0.96</td>
<td>0.96</td>
<td>0.01</td>
</tr>
<tr>
<td>MPEG-4 (a)</td>
<td>0</td>
<td>0</td>
<td>0.04</td>
<td>0</td>
</tr>
<tr>
<td>MPEG-4 (b)</td>
<td>0.02</td>
<td>0.02</td>
<td>0.06</td>
<td>0.01</td>
</tr>
<tr>
<td>AVC (a)</td>
<td>0.01</td>
<td>0</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td>AVC (b)</td>
<td>0</td>
<td>0.01</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td>AVC (c)</td>
<td>0.06</td>
<td>0.06</td>
<td>0.15</td>
<td>0.01</td>
</tr>
<tr>
<td>DIRAC</td>
<td>0</td>
<td>0.02</td>
<td>0</td>
<td>0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>c_2</th>
<th>MPEG-2</th>
<th>MPEG-4</th>
<th>AVC</th>
<th>MPEG-2</th>
<th>MPEG-4</th>
<th>AVC</th>
<th>MPEG-2</th>
<th>MPEG-4</th>
<th>AVC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPG-2</td>
<td>0.94</td>
<td>0.96</td>
<td>0.96</td>
<td>0.05</td>
<td>0.04</td>
<td>0.03</td>
<td>0</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td>MPEG-4 (a)</td>
<td>0</td>
<td>0</td>
<td>0.04</td>
<td>0.93</td>
<td>0.92</td>
<td>0.76</td>
<td>0.02</td>
<td>0.02</td>
<td>0.2</td>
</tr>
<tr>
<td>MPEG-4 (b)</td>
<td>0.02</td>
<td>0.02</td>
<td>0.06</td>
<td>0.87</td>
<td>0.87</td>
<td>0.69</td>
<td>0.06</td>
<td>0.05</td>
<td>0.25</td>
</tr>
<tr>
<td>AVC (a)</td>
<td>0.01</td>
<td>0</td>
<td>0</td>
<td>0.14</td>
<td>0.24</td>
<td>0.06</td>
<td>0.79</td>
<td>0.68</td>
<td>0.94</td>
</tr>
<tr>
<td>AVC (b)</td>
<td>0</td>
<td>0.01</td>
<td>0</td>
<td>0.13</td>
<td>0.2</td>
<td>0.05</td>
<td>0.81</td>
<td>0.69</td>
<td>0.94</td>
</tr>
<tr>
<td>AVC (c)</td>
<td>0.06</td>
<td>0.06</td>
<td>0.15</td>
<td>0</td>
<td>0.03</td>
<td>0</td>
<td>0.92</td>
<td>0.87</td>
<td>0.81</td>
</tr>
<tr>
<td>DIRAC</td>
<td>0</td>
<td>0.02</td>
<td>0</td>
<td>0.09</td>
<td>0.12</td>
<td>0.01</td>
<td>0.13</td>
<td>0.12</td>
<td>0.21</td>
</tr>
</tbody>
</table>
Are **codec** and **quality** coherent in time?

[Verde et al. ICIP 2018]
Coding-based footprints: temporal coherence

Problem
• Given a decoded video sequence, detect whether it is a compilation from multiple video shots.

Assumptions
• Shots are seldom originally encoded with the exact same codec or parameters due to different sources and used software.
Coding-based footprints: temporal coherence

Main Pipeline
- Compute a frame-wise indicator of the used codec
- Compute a frame-wise indicator of the video quality
- Check inconsistency of these indicators frame-by-frame
Feature Extraction

Codec Features
- A CNN is trained to classify 4 different codecs (MPEG2, MPEG4, H264, H265)
- Feature vector is $f_C(n) = [f_{H264}^p(n), f_{H265}^p(n), f_{MPEG2}^p(n), f_{MPEG4}^p(n)]$

Quality Features
- A CNN is trained to classify 4 different qualities (H264 with QP=5, 10, 15, 20)
- Feature vector is $f_Q(n) = [f_{low}^p(n), f_{m-low}^p(n), f_{m-high}^p(n), f_{high}^p(n)]$
Coding-based footprints: temporal coherence

Temporal inconsistency analysis

**Feature Merge**
- Feature vectors are concatenated into a single one
  \[ f_{CQ}(n) = [f_C(n), f_Q(n)] \]

**Time Analysis**
- Compute MSE between feature pairs
  \[ \Delta f_{CQ}(n) = \text{MSE}(f_{CQ}(n), f_{CQ}(n + 1)) \]

**Threshold MSE**
Coden-based footprints: temporal coherence

Visual example

![Graphs showing temporal coherence](https://media.xiph.org/video/derf/)

(a) Pristine
(b) Composition
Coding-based footprints: temporal coherence

Challenging Example
Coding-based footprints: temporal coherence

Challenging Example
Coding-based footprints: temporal coherence

### Video Codec Identification Results

<table>
<thead>
<tr>
<th></th>
<th>H264</th>
<th>H265</th>
<th>MPEG2</th>
<th>MPEG4</th>
</tr>
</thead>
<tbody>
<tr>
<td>H264</td>
<td>0.76</td>
<td>0.20</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>H265</td>
<td>0.12</td>
<td>0.85</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>MPEG2</td>
<td>0.02</td>
<td>0.01</td>
<td>0.84</td>
<td>0.13</td>
</tr>
<tr>
<td>MPEG4</td>
<td>0.05</td>
<td>0.03</td>
<td>0.03</td>
<td>0.89</td>
</tr>
</tbody>
</table>

### Video Quality Identification Results

<table>
<thead>
<tr>
<th></th>
<th>high</th>
<th>m-high</th>
<th>low</th>
</tr>
</thead>
<tbody>
<tr>
<td>high</td>
<td>0.84</td>
<td>0.12</td>
<td>0.00</td>
</tr>
<tr>
<td>m-high</td>
<td>0.09</td>
<td>0.83</td>
<td>0.00</td>
</tr>
<tr>
<td>m-low</td>
<td>0.03</td>
<td>0.07</td>
<td>0.85</td>
</tr>
<tr>
<td>low</td>
<td>0.00</td>
<td>0.00</td>
<td>0.96</td>
</tr>
</tbody>
</table>
Splicing Detection Results

![ROC Curve for Splicing Detection](image)

- Δf_Q: AUC = 0.86
- Δf_C: AUC = 0.93
- Δf_Q: AUC = 0.96
Paolo Bestagini

Analysis on a single video

Coding-based
footprints

Number of
compressions

Type of
codec

Coding-based

footprints

Image splicing
Analysis on a single video

- **Coding-based footprints**
  - Image splicing

- **Editing-based footprints**
  - Video copy-paste
  - Video copy-move

- **Acquisition-based footprints**
  - Re-capture

- **Type of codec**
  - Image splicing
  - Video copy-paste
  - Video copy-move
  - Re-capture
Editing-based footprints
Editing-based footprints
Editing-based footprints
Editing-based footprints
Editing-based footprints

Forged

Original
Editing-based footprints

Forged

Original
Is the video forged?

[Bestagini et al. ICASSP 2013]
[Bestagini et al. MMSP 2013]
Editing-based footprints

- Video *forgeries* can be operated through different kinds of editing attacks

- We considered:
  1. insertion of a *still image*
  2. insertion of a portion of *video from the same source*
  3. insertion of a portion of *video from a different source*
Editing-based footprints: image copy-paste

• Problem:
  • An image is inserted and repeated in time

• Method:
  • Exploit characteristic residual between adjacent frames

![Image](image.png)

(a) Original residual  (b) Tampered residual  (c) Tampered and compressed residual
Editing-based footprints: image copy-paste

ALGORITHM:

- Define the residual $r_{i,j}^t = x_{i,j}^t - x_{i,j}^{t+1}$
  - Zero for possibly tampered pixels
- Define the residual mask as $m_{i,j}^t = \begin{cases} 
1 & \text{if } r_{i,j}^t = 0, \\
0 & \text{otherwise}, 
\end{cases}$
- Apply an erosion with a Structuring Element $H^{d_i,d_j,d_t}$ and obtain $E = \{e_{i,j}^t\} = M \ominus H^{d_i,d_j,d_t}$
  - Remove small areas
- Compute the feature vector $F_{i,j} = [f_{i,j}^1, f_{i,j}^2]$
  - $f_{i,j}^1$: cardinality of the longest set of ones in $(i,j)$
  - $f_{i,j}^2$: starting $t$ value of the longest set of ones
- Search the longest set of ones starting from the same $t$
Editing-based footprints: image copy-paste

(a) Tampered frame

(b) Detected mask

(c) $f_{i,j}^1$

(d) $f_{i,j}^2$
Editing-based footprints: image copy-paste

• Problem:
  • A video is inserted from the same sequence

Method:
  • Implementation of an automatic correlation analysis to detect local duplication
Editing-based footprints: image copy-paste

• Problem:
  • A video is inserted from the same sequence

Method:
• Implementation of an automatic correlation analysis to detect local duplication
Editing-based footprints: image copy-paste

Algorithm:

- Compute the residual $R = \{r_{i,j}^t\}$
- Divide the residual into non-overlapping 3D blocks $B_m^n$
- Compute the phase correlation
  \[
  C_{i,j}^t(B_m^n) = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(B_m^n)\mathcal{F}(R)^*}{|\mathcal{F}(B_m^n)\mathcal{F}(R)^*|}\right)
  \]
- Compute the maximum correlation value for each time position $c_{B_m}^t = \max_{i,j}(|C_{i,j}^t(B_m^n)|)$
- Search for peaks indicating duplication by thresholding the max-mean ratio
  \[
  p_{B_m}^n = \frac{\max(c_{B_m}^t)}{\frac{1}{(T-1)} \sum_t c_{B_m}^t}.
  \]
- Check if the detected duplicated block is similar to its original version (MSE)
Editing-based footprints: image copy-paste

![Original](original.png)

![Forged](forged.png)

![Detected Duplication](detected_duplication.png)

original  forged  detected duplication
Editing-based footprints: image copy-paste

• **Problem:**
  - A video is inserted from the same a different sequence at with different frame-rate

• **Method:**
  - Search for traces left by frame-rate equalisation
  - Up-sampling and down-sampling leave a characteristic pixel correlation in time

\[
X_{ij}^\omega(\omega t)
\]
Algorithm:

- Estimate each frame from their neighbors
  - Compute motion vectors
  - Average frames

- Compute the prediction error
  - Original frames $\Rightarrow$ high error
  - Predicted frames $\Rightarrow$ low error

$$e_{ij}(\omega t) = X_{ij}^\omega(\omega t) - \sum_{k=-K}^{K} h_k^* \cdot X_{m_t, i, j}^{\omega} n_{t, i, j}(\omega t + \omega k).$$

$$e(\omega t) = \sum_{ij} |e_{ij}(\omega t)|^2$$

- Estimate error periodicity (spectral analysis)
  - If non-periodic $\Rightarrow$ not interpolated
  - If periodic $\Rightarrow$ interpolated
Editing-based footprints: image copy-paste

Foreman (original)

Foreman (interpolated)

Fps: 30 ➞ 90

Foreman (interpolated)

Spectral analysis

\[ \Delta f = 0.5 - |\omega - 0.5|. \]
Paolo Bestagini

Work organisation

Coding-based footprints

Editing-based footprints

Acquisition-based footprints

Type of codec

Image splicing

Video copy-paste

Video copy-move

Video copy-move

Re-capture

Acquisition footprints

Editing-based footprints

Image splicing

Video copy-paste
Re-capture

Acquisition-based footprints
Acquisition-based footprints
Acquisition-based footprints
Acquisition-based footprints

- Re-acquisition is a powerful anti-forensic tool

- Re-acquired videos are visually similar to the originals

- Many footprints are masked
  - Detectors can be fooled

![Histograms of DCT coefficients](image)

- Original: DCT statistics
- Recaptured: DCT statistics
Is the video recaptured?
Acquisition-based footprints: re-capture

• Setup:
  • A video is re-captured from a LCD monitor

• Ghosting as filtering:

\[ X_1 = \text{Orig. scene} \]

\[ X_2 = \text{Orig. scene} \]

\[ Y_1 = \text{Recap. scene} \]

\[ \alpha \]

\[ + (1 - \alpha) \]

Approximating the motion as a translation between adjacent frames:

\[ X_1 = \text{Orig. scene} \]

\[ H(\alpha) = \text{Ghost. Filter} \]

\[ Y_1 = \text{Recap. scene} \]

\[ \ast \]

\[ \alpha \]

\[ (1 - \alpha) \]
Acquisition-based footprints: re-capture

- Filter shape is derived from motion estimation
Acquisition-based footprints: re-capture

• **Method:**
  • Minimize a cost function to detect whether key-points underwent “ghosting filtering”

• **Results:**
  • Detection accuracy over 91%
From **which camera** does the video come from?
Acquisition-based footprints: camera attribution

- Photo Response Non Uniformity:
  - It enables linking images to devices

- How to:
  - Extract noise pattern from images
  - Compute correlation
Acquisition-based footprints: camera attribution

• Application:
  • Video compilation detection and segmentation

• Method:
  • Compute cumulative correlation \( c(f) = \rho(\mathbf{W}_r, \mathbf{W}(f)\mathbf{I}_r) \)

![Diagram of video compilation with shots labeled S1, S2, S3, and frame indices for each shot]
Acquisition-based footprints: camera attribution

- **Challenges:**
  - Aggressive coding
  - Digital video stabilisation
Video

copy-move

Re-capture

Acquisition-based
footprints

Editing-based
footprints
Work organisation

- **Number of compressions**
- **Type of codec**
- **Coding-based footprints**
- **Video splicing**
- **Video copy-paste**
- **Video copy-move**
- **Re-capture**
- **Editing-based footprints**
- **Acquisition-based footprints**
- **Video forensics**
Put everything together

- **input video**
  - Re-capture: yes → Image splicing
  - Re-capture: no → Video copy-paste
  - Re-capture: no → Video copy-move

- Video copy-paste: yes → Number of compressions
  - Number of compressions: many → Re-capture: yes
    - Re-capture: yes → Type of codec
  - Number of compressions: 1 → Re-capture: no
  - Number of compressions: no → Re-capture: no

Type of codec

- Image splicing
- Video copy-paste
- Video copy-move
From one to many

- **Coding-based footprints**
  - Number of compressions
  - Type of codec

- **Image splicing**

- **Editing-based footprints**

- **Video copy-paste**

- **Video copy-move**

- **Re-capture**

- **Acquisition-based footprints**

- **Video forensics**

From one to many

- Coding-based footprints
- Editing-based footprints
- Acquisition-based footprints

Video forensics

- Number of compressions
- Type of codec
- Image splicing
- Video copy-paste
- Video copy-move
- Re-capture

Video forensics
From one to many
From one to many
Applications
Coding-based applications

• Which video has been more processed?
  • Extend Benford’s law to base-N first digits
  • Fit logarithmic curve
  • Check goodness of fit (processing age)
  • The better, the younger!

(a) $N_c = 1$ QP=21.
(b) $N_c = 2$ QP=25.
(c) $N_c = 3$ QP=28.
(d) $N_c = 4$ QP=24.

[Milani et al. EUSIPCO 2017]
Acquisition-based applications

- Which views are the redundant?
  - match video PRNUs to detect those from same device
Parent reconstruction

• Who is my parent?
• It is possible that we are analysing a short shot (child) of a longer sequence (parent)
  • e.g., a VIP speech

• Can we find other (partially overlapping) child sequences to reconstruct the parent?
Parent reconstruction

• Download a set of videos related to the topic under analysis:
Parent reconstruction

• Download a set of videos related to the topic under analysis:
Parent reconstruction

- Analyse each pair of sequences exploiting a robust hash algorithm
  - A sequence is split in overlapping time segments of 64 frames each
Parent reconstruction

• Analyse each pair of sequences exploiting a robust hash algorithm
  • each block is described by a binary hash

- Every frame in the block is spatially resized to 32x32 pixels
  • The block now measures 32x32x64 pixels
- 3D DCT is applied to the block
- 64 DCT coefficients are selected
- This 64 DCT coefficients are binarized according to their median value
  • 32 are set to zero, 32 are set to 1
- The hash is this 64 binary string
Parent reconstruction

- Analyse each pair of sequences exploiting a robust hash algorithm
  - Hashes of different blocks are compared by computing hamming distance
Parent reconstruction

- Compute the **distance between every block** of sequence 1 and every block of sequence 2

- **Non-near duplicates**
  - High distance
  - No regular patterns

- **Near duplicates**
  - Low distances = matching
  - Start and end points used for alignment
Parent reconstruction

• Analyse each pair of sequences exploiting a robust hash algorithm:
Parent reconstruction

- Analyse each pair of sequences exploiting a robust hash algorithm:
Parent reconstruction

1. [Image of a 2D plot with axes labeled $n_1$ and $n_2$.]
2. [Image of another 2D plot with axes labeled $n_1$ and $n_2$.]
3. [Image of a 2D plot with axes labeled $n_1$ and $n_2$.]
4. [Image of a 2D plot with axes labeled $n_1$ and $n_2$.]
Parent reconstruction

• Segment each sequence according to the matching shots:
Parent reconstruction

• Segment each sequence according to the matching shots:
Parent reconstruction

• Reconstruct the most part of the parent sequence for the analysis:
Parent reconstruction

• Reconstruct the most part of the parent sequence for the analysis:
Parent reconstruction

- Being able to **reconstruct the parent** from the children enables to shed very interesting insights on the way content is reused:

  1. Analyse the **context** from which a child sequence was taken

  2. **Reconstruct** sequences no longer available online in their totality

  3. Establish causal **relationship** between children
Conclusions

• Remarks
  • Forensics vs. Anti-forensics
  • Single video analysis is just part of the problem
  • Multiple video analysis paves the way to the development of novel applications

• Open questions
  • Merge results from content- and context-aware detectors
    • Do metadata match the video content?
  • Deal with big data
    • Time-consuming algorithms need optimisation
  • Deep learning
    • Still under-investigated in video forensics (space-time?)
    • Training data hardly available…
References

• S. Verde, L. Bondi, P. Bestagini, S. Milani, G. Calvagno, S. Tubaro, “Video Codec Forensics Based on Convolutional Neural Networks”, IEEE International Conference on Image Processing (ICIP), Athens, Greece, 2018


References


Thank you for the attention!

Any questions?