



UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

SCUOLA DI INGEGNERIA  
Corso di Laurea Magistrale in Ingegneria  
Informatica

# Forensic Analysis of Video File Containers

**Saverio Meucci**

*Relatori:* Prof. Alessandro Piva, Prof. Fabrizio Argenti

*Co-Relatori:* Ing. Marco Fontani, Dott. Massimo Iuliani

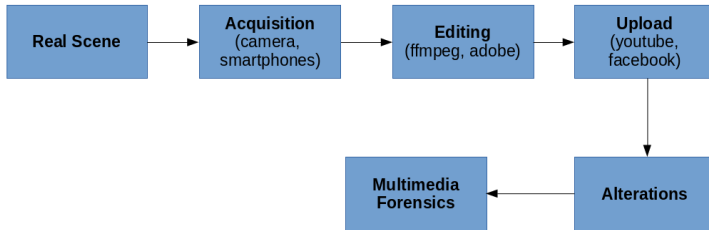
ANNO ACCADEMICO 2015/2016

# Introduction

---

In an increasingly digital world, the analysis of multimedia objects is rapidly assuming importance in the context of **digital investigation**.

Multimedia Forensics has developed many techniques with the goal of providing aid in making decisions about a digital content **authenticity**, **integrity** and **origin**.



# Introduction

---

In this regard, techniques mainly use two different approaches:

1. **Audio-video signal:** the research of inconsistencies and artefacts in the digital content.
2. **Metadata/Container:** the determination of their compatibility, completeness, and consistency.

# Video File Container - What's inside?

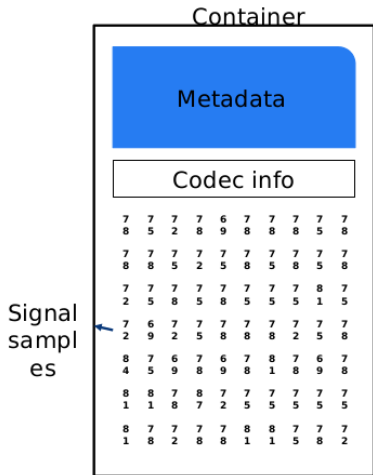
**Container data**, structured information about the content:

- Content-related metadata
- Number of tracks/signals

**Codec data**, necessary information to decode and present the signal:

- Quantization tables.
- Information for entropy decoding

**Encoded signal(s)**

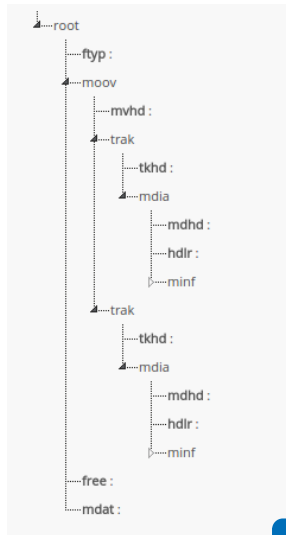


# Video File Container - Structure

As defined by the *ISO Base Media File Format Standard* [1], file containers have a **object-oriented** type structure.

Each object, called **box** or **atom**, has an identifier and includes specific information about the media.

Boxes can have **attributes** and can contain other boxes.



## Video File Container - Why?

---

As noticed by *Gloe et al.* [2], the video format standards (e.g. *ISO Base Media* [1], *MP4* [3], *MOV* [4]) for the file container prescribe only a limited number of mandatory features.

## Video File Container - Why?

---

As noticed by *Gloe et al.* [2], the video format standards (e.g. *ISO Base Media* [1], *MP4* [3], *MOV* [4]) for the file container prescribe only a limited number of mandatory features.



**Freedom of interpretation** for the device manufacturers in terms of design decisions (order of the boxes, attributes values, etc.).

## Video File Container - Why?

---

As noticed by *Gloe et al.* [2], the video format standards (e.g. *ISO Base Media* [1], *MP4* [3], *MOV* [4]) for the file container prescribe only a limited number of mandatory features.



**Freedom of interpretation** for the device manufacturers in terms of design decisions (order of the boxes, attributes values, etc.).



Low-level information that we have exploited with regard to **Source Identification** and **Integrity Verification**.



## Source Identification

---

Given a video, we want to assess its **origin** based on its file container.

We split the problem in **binary questions**.

Ex. Does the video belongs to Samsung?

... to Samsung Galaxy S3?

... to Huawei G6?

... to Apple?

... to Apple iPhone 5?

Given a question, a training dataset is queried to obtain two classes (videos for which the answer is true, and the complementary).

For each question, we want to define a compatibility score.

## Source Identification - Training

---

Determine whether a video belongs to a class  $C$  (e.g. Samsung).

We split the ground-truth in two sets:

$$\Omega = X_C \cup X_{\overline{C}} = x_1, \dots, x_{N_C} \in C \cup y_1, \dots, y_{N_{\overline{C}}} \in \overline{C}$$

$\Omega$  contains all the attributes  $\omega$  of the boxes contained in each of the ground-truth media  $X$ .

## Source Identification - Training

---

Determine whether a video belongs to a class  $C$  (e.g. Samsung).

We split the ground-truth in two sets:

$$\Omega = X_C \cup X_{\bar{C}} = x_1, \dots, x_{N_C} \in C \cup y_1, \dots, y_{N_{\bar{C}}} \in \bar{C}$$

$\Omega$  contains all the attributes  $\omega$  of the boxes contained in each of the ground-truth media  $X$ .



We determine the **discrimination power** of each of the attributes  $\omega$  for the class  $C$  and  $\bar{C}$ .

$$W_C(\omega) = \frac{\sum_{i=1}^{N_C} |X_i \cap \omega|}{N_C}$$

$$W_{\bar{C}}(\omega) = \frac{\sum_{i=1}^{N_{\bar{C}}} |X_i \cap \omega|}{N_{\bar{C}}}$$

## Source Identification - Test

---

Given a media query  $X = \omega_1, \dots, \omega_t$ , we solve the two hypothesis test problem:

$$H_0 : X \in \overline{C}$$

$$H_1 : X \in C$$

To do so, we determine the **likelihood** ratio of observing  $\omega_j, j = 1 \dots t$ .

$$P(\omega_j | H_0) = W_{\overline{C}}(\omega_j)$$

$$P(\omega_j | H_1) = W_C(\omega_j)$$

$$L(X) = \prod_{\omega_j} \frac{W_C(\omega_j)}{W_{\overline{C}}(\omega_j)}$$

Then,  $l(X) = \ln L(X)$  can be used to determine whether  $X$  belongs to class  $C$ .

## Source Identification - Correlated Features

---

Some features might be correlated. For each box, we consider the **entropy** of its attributes in order to remove redundant information.

When considering a box, given a vector of likelihood ratios  $\bar{x} = (x_1, \dots, x_n)$ , we compute the likelihood for that box as:

$$L(\bar{x}) = \prod_{i=1}^n x_i^{\alpha_i(\gamma_i)}$$

with

$$\gamma_i = -\frac{n}{\log n} P(x_i) \log P(x_i)$$

and where  $P(x_i)$  represents the probability of finding that value of ratio in the vector.

## Source Identification - Dataset

---

The dataset is composed of 260 videos acquired from smartphones and tablets with Android (Samsung, Huawei) and iOS.

OS	BRAND	MODEL	#
Android			150
	Samsung		132
		Galaxy S3	18
		Galaxy S3 mini	36
		Galaxy S4 mini	18
		Galaxy Tab 3	36
		Galaxy Tab A	9
		Galaxy Trend Plus	15
	Huawei		18
		G6	18
iOS			110
	Apple		110
		iPad 2	15
		iPad mini	15
		iPhone 4S	14
		iPhone 5C	18
		iPhone 5	31
		iPhone 6	17
			260

## Source Identification - Experiments

---

The tests are divided in two types, changing the definition of a class of device:

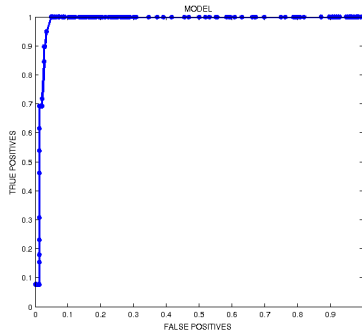
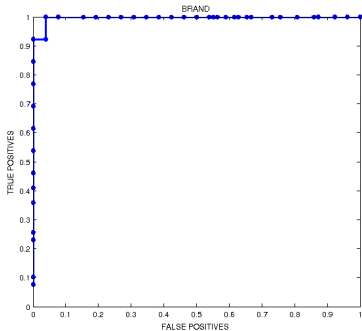
- **Brand:** we try to identify the test videos brand (3 brands).
- **Model:** we try to identify both brand and model (13 models).

For each of these types, we consider:

- **Binary Classification:** for each class of devices in the dataset, we try to correctly classify the test videos.
- **Retrieval:** how many times the correct classes are in the first position, in the top three position, or in the top five position (ordered by the likelihood ratios).

# Source Identification - Results

Type	ACC	THRESHOLD	TOP 1	TOP 3	TOP 5
Brand	98.08%	0	92%	-	-
Model	97.54%	3.5	84.62%	100%	100%





## Integrity Verification

---

Given a query video  $X$  that supposedly comes from a certain device.

## Integrity Verification

---

Given a query video  $X$  that supposedly comes from a certain device.



We want to assess if this supposition is true or if the video has been altered in some way.

## Integrity Verification

---

Given a query video  $X$  that supposedly comes from a certain device.



We want to assess if this supposition is true or if the video has been altered in some way.



We obtain a reference video  $Y$ , acquired by the supposed device.

## Integrity Verification

---

Given a query video  $X$  that supposedly comes from a certain device.



We want to assess if this supposition is true or if the video has been altered in some way.



We obtain a reference video  $Y$ , acquired by the supposed device.



By comparing the two file containers, we compute the percentage of differences.

## Integrity Verification - Experiments

---

For these experiments, we have altered the videos of the dataset with different tools:

- **Ffmpeg**: we have directly cut the videos, without re-encoding.
- **Exiftool**: we have changed the metadata related to Date and Time.
- **YouTube**: we have uploaded and downloaded the videos from *YouTube*.

Using their file containers, we compute the differences:

1.  $(x_1, \dots, x_n) \in C_i, (x_i, x_j) \rightarrow d_{ij}$
2.  $(\overline{x_1}, \dots, \overline{x_n}) \in \overline{C_i}, (x_i, \overline{x_j}) \rightarrow \overline{d_{ij}}$

## Integrity Verification - Results

---

N.	Tool	ACC	THRESHOLD
1	Ffmpeg	100%	0.385
2	Exiftool	100%	0.001
3	YouTube	100%	0.470

We were always able to correctly separate the original videos from the modified ones.

These tools alter the original file container.

# Web Application

The screenshot displays a web application interface for Source Identification. At the top, a dark navigation bar contains the tabs 'Classify', 'Compare', 'Test', and 'About'. The main content area is organized into three vertical panels. The 'Class' panel on the left features a 'Brand' dropdown menu set to 'apple' and a 'Model' dropdown menu with a list of options including 'Any', 'ipad2', 'ipadmini', 'iphone4s', 'iphone5', 'iphone5c', and 'iphone6'. The 'Upload' panel in the center has a 'Select Video/XML:' section with an 'Upload' button and the text 'No file chosen', and a 'Download vft-parse.jar:' section with a 'Download' button. The 'Output' panel on the right contains a large, empty text area with a vertical scrollbar and a red 'Run query' button at the bottom right.

Figura : Interface for the Source Identification feature.

## Conclusions

---

- Using the video file containers, we implemented two approaches for Source Identification and Integrity Verification.
- Video file container turned out to be a powerful tool; both approaches achieved promising results.
- Should be considered preliminary work; further developments:
  - Perform tests with a higher variety of devices.
  - Take into consideration the version of the operating system.
  - Specialize how the attributes are compared (e.g. check for format for Date and Time).



## References

---

- [1] I. 14496. *Information Technology. Coding of audio-visual objects, part 12: ISO Base Media File Format*, 3rd ed. 2008.
- [2] T. Gloe, A. Fischer, and M. Kirchner. Forensic analysis of video file formats. *Digital Investigation*, 11, Supplement 1:S68-S76, 2014. Proceedings of the First Annual DFRWS Europe.
- [3] I. 14496. *Information Technology. Coding of audio-visual objects, part 14: Mp4 File Format*, 2003.
- [4] I. Apple Computer. *Quicktime file format*, 2001.