

Thực hành tách tin trong miền tần số của video

Tách tin từ hệ số DC và AC

Nội dung và hướng dẫn bài thực hành

Mục đích

Hướng dẫn sinh viên các bước cơ bản trong quy trình tách tin từ hệ số DC và AC trong miền tần số của các video theo chuẩn định dạng .mpeg

Cung cấp kiến thức cho sinh viên về cách video định dạng .mpeg lưu trữ các frame, các loại khung; I, P, B.

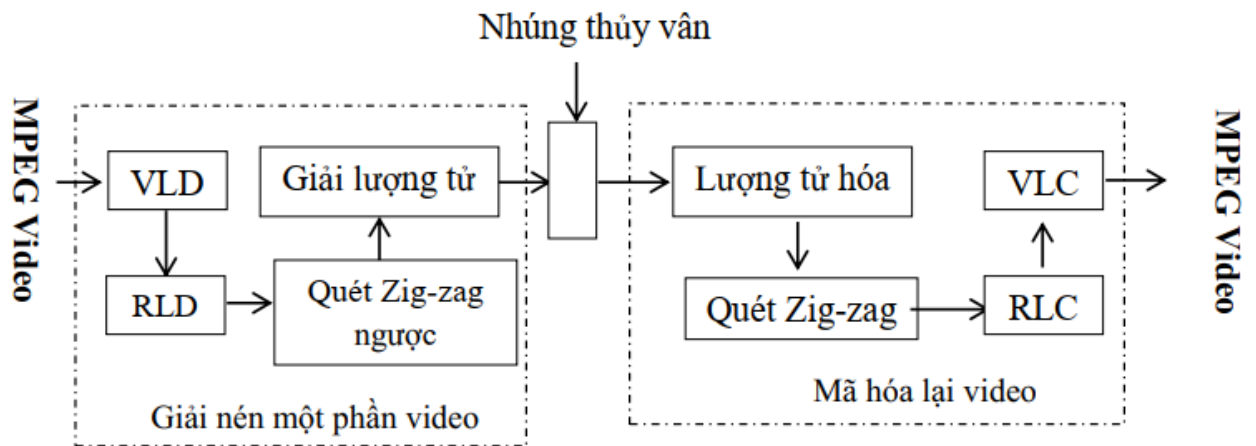
Nâng cao nhận thức chung về các kĩ thuật tách tin trong bối cảnh giấu tin chứa nhiều ứng dụng nhưng cũng có nhiều thách thức về mặt an toàn, phương pháp.

Yêu cầu đối với sinh viên

Có kiến thức cơ bản về hệ điều hành linux, ubuntu. Công cụ ffmpeg.

Biết lập trình python cơ bản.

Hiểu được kiến thức nền tảng trong lĩnh vực giấu - tách tin.



Định dạng video .mpeg có 3 loại khung

1. I-frame (Intra-coded frame)

- Còn gọi là Intra frame.
- Là khung tự mã hóa độc lập, không phụ thuộc vào bất kỳ khung nào khác.
- Giống như một "bức ảnh hoàn chỉnh".
- Dùng để tái lập điểm tham chiếu (ví dụ như lúc tua video).

- Do chứa đầy đủ thông tin hình ảnh, nên kích thước I-frame thường lớn nhất.
- 2. P-frame (Predictive-coded frame)
 - Còn gọi là Predicted frame.
 - Mã hóa dựa trên dữ liệu của I-frame hoặc một P-frame trước đó.
 - Chỉ lưu sự thay đổi so với khung trước, nên kích thước nhỏ hơn I-frame.
 - P-frame giúp giảm rất nhiều dung lượng video.
- 3. B-frame (Bidirectionally-predictive-coded frame)
 - Còn gọi là Bidirectional frame.
 - Mã hóa dựa trên cả khung trước và khung sau (I hoặc P).
 - Nó có thể tham chiếu 2 hướng, nên khả năng nén hiệu quả nhất.
 - B-frame cần nhiều tính toán hơn để mã hóa/giải mã nhưng cực kỳ tiết kiệm dung lượng.

Nội dung thực hành

Tải nội dung lab, chạy lệnh:

```
imodule https://github.com/TheBoyWhoDev/dct-video-extract/raw/main/imodule.tar
```

Khởi động bài lab:

```
labtainer -r dct-video-extract
```

Sau khi khởi động bài lab, một terminal ảo sẽ xuất hiện, tại thư mục Home của người dùng sẽ có các file với các nhiệm vụ sau:

- *video.mp4*: file video có chứa tin giấu.

- *extract.py*: file python chứa code tách tin từ 1 ảnh, cú pháp chạy lệnh:

```
python3 extract.py <ImagePath>
```

<ImagePath>: đường dẫn đến file ảnh chứa tin giấu

- *batch_extract.sh*: file shell script gọi lệnh công cụ *extract.py* để tách tin từ tất cả file ảnh trong thư mục hiện tại (trong bài thực hành này, ảnh là các frame được tách ra từ *video.mp4*). Cú pháp chạy lệnh:

```
./batch_extract.sh
```

Nhiệm vụ cần làm:

Nhiệm vụ 1: Tách 1 số frame từ video chứa tin giấu

Chạy lệnh ls để kiểm tra các file & thư mục được cung cấp cho bài lab

lệnh sau sẽ tách 4 frame từ video.mp4 và lưu vào thư mục hiện hành với định dạng tên file frame_%03d.png, ví dụ frame_001.png

```
ffmpeg -i video.mp4 -vf "select=lte(n\,3)" -vsync vfr frame_%03d.png
```

Nhiệm vụ 2: Thử tách tin từ 1 frame duy nhất

Để hiểu rõ hơn về logic tách tin sinh viên có thể chạy lệnh, nano extract.py để nghiên cứu cách mã nguồn hoạt động.

Chạy lệnh:

```
python3 extract.py frame_003.png
```

Lệnh trên sẽ tách tin từ 1 file ảnh, ở đây là frame_003.png

Nhiệm vụ 3: Thực hiện tách tin từ các frame đã lấy từ video

```
chmod +x batch_extract.sh
```

```
// cấp quyền thực thi chmod +x batch_extract.sh
```

```
./batch_extract.sh
```

```
// thực hiện tách tin từ các frame đã tách từ video.mp4 (4 frame đã tách ở nhiệm vụ 1)
```

Kết quả tách tin được lưu trong file result.txt, tin giấu trong mỗi frame sẽ được lưu trên mỗi dòng riêng biệt.

Nhiệm vụ 4: Quan sát kết quả tách tin, tìm ra thông điệp bí mật

Thực hiện lệnh sau để in ra màn hình chuỗi văn bản được giấu trong các frame

```
cat result.txt
```

Kết thúc bài lab:

Lệnh checkwork:

```
checkwork
```

Lệnh dừng lab:

```
stoplab
```

Khởi động lại bài lab:

```
labtainer -r dct-video-extract
```