

Contents

1	TVID projet	1
1.1	Auteurs	1
1.2	A. Jouer un flux MPEG-2 élémentaire de test	1
1.3	B. Jouer un flux vidéo de chaîne d'infos américaine assez notoire	4
1.4	C. Jouer un flux vidéo de chaînes de divertissement asiatiques:	5

1 TVID projet

1.1 Auteurs

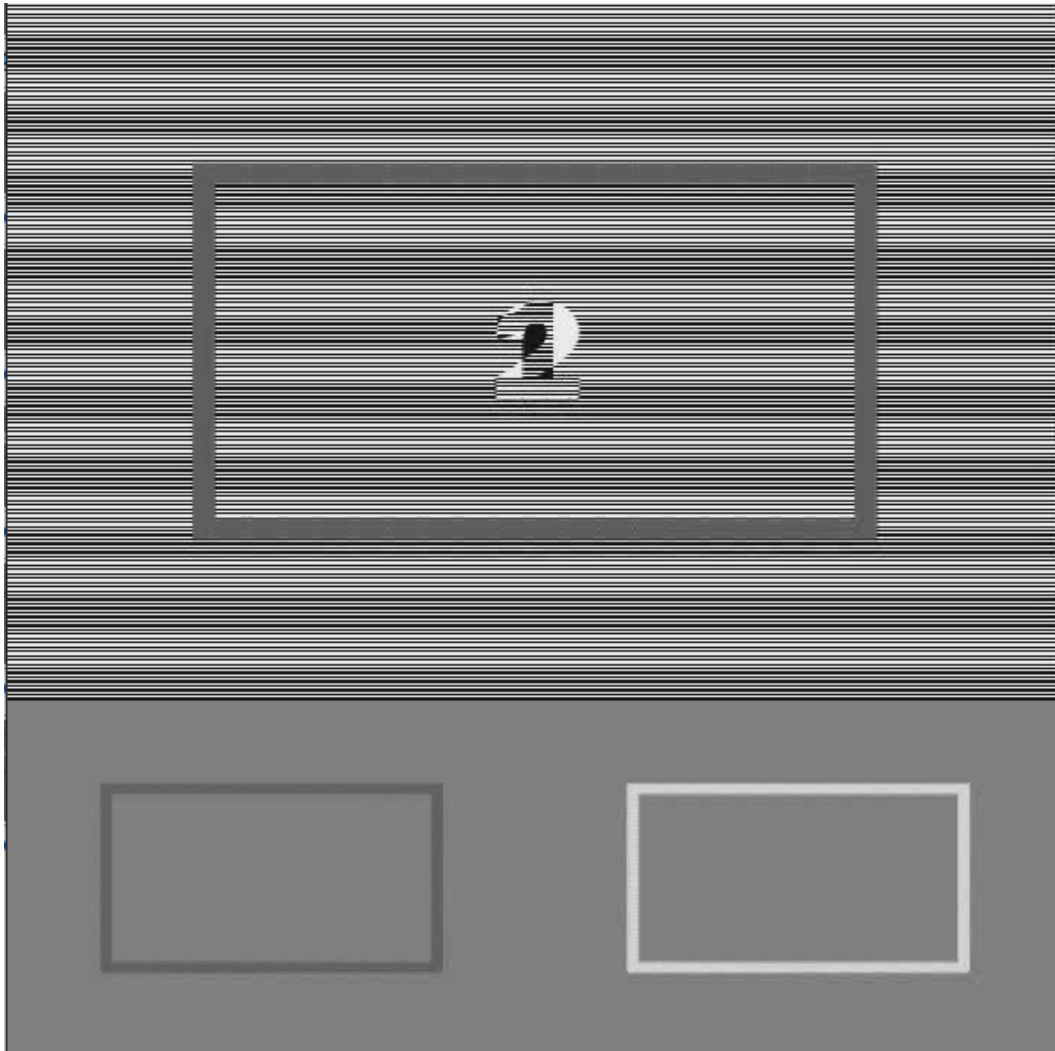
- Ilan Guenet (ilan.guenet)
- Quentin Kaci (quentin.kaci)

1.2 A. Jouer un flux MPEG-2 élémentaire de test

1. Dans le dossier videos/elementary, avec vlc, visualisez les séquences MPEG-2 suivantes : `vlc -V <renderer> bw_numbers.m2v / pendulum.m2v / Jaggies1.m2v` avec `<renderer> = x11` ou `xvideo`, selon ce qui fonctionne sur votre machine

`vlc -V x11 bw_numbers.m2v` Pour dentrelacer, clic droit -> video-> deintrelace -> Off\On

2. Avec mpeg2dec, convertissez en pile d'images votre séquence MPEG-2 choisie (cf. aide de mpeg2dec). Il est recommandé de sortir les images au format pgm pour pouvoir les traiter ultérieurement.
`../tools/mpeg2dec/src/mpeg2dec -o pgm ../videos/elementary/bw_numbers.m2v`
3. Observez les pgm générées. Comment sont-elles structurées? Quel est le format de l'image: résolution, profondeur, sampling mode?



- Format: PGM
 - image: 2 images consécutives entrelacées dans une même image (top first field).
 - Résolution: 720x480
 - chroma: 360x240
 - sampling mode: 4:2:0 (YUV)
 - Profondeur: 8-bit
4. Modifiez mpeg2dec pour logger simplement les flags `progressive_frame`, `top_field_first`, `repeat_first_field` de chaque image décodée.

Dans le fichier `mpeg2dec.c:55` `static int verbose = 1;`, mettre le `verbose` à 1.

- Progressive Frame: **PROG**
- Top Field First: **TFF**
- Repeat First Field: **RFF** Log:

```

5c S- --- --- - SEQUENCE MPEG2 MP@ML 720x480 chroma 360x240 fps 29.97 maxBps 30000
      64 SG --- --- - GOP CLOSED 0: 0: 0: 0
      84 SG AA- --- ? PICTURE I fields 2 TFF time_ref 2 offset 0/0 0/0
5084 SG aa- --- ? SLICE
50c4 SG BB- BB- B PICTURE B fields 2 TFF time_ref 0 offset 0/0 0/0
6c84 SG bb- bb- b SLICE
6cc4 SG CC- CC- C PICTURE B fields 2 TFF time_ref 1 offset 0/0 0/0
7804 SG cc- cc- c SLICE
7844 SG DD- aa- ? PICTURE P fields 2 TFF time_ref 5 offset 0/0 0/0
8ec4 SG dd- aa- ? SLICE
8f04 SG EE- EE- E PICTURE B fields 2 TFF time_ref 3 offset 0/0 0/0
9984 SG ee- ee- e SLICE
99c4 SG FF- FF- F PICTURE B fields 2 TFF time_ref 4 offset 0/0 0/0
a3c4 SG ff- ff- f SLICE
a404 SG GG- dd- a PICTURE P fields 2 TFF time_ref 8 offset 0/0 0/0
c244 SG gg- dd- a SLICE
c284 SG HH- HH- H PICTURE B fields 2 TFF time_ref 6 offset 0/0 0/0
e1c4 SG hh- hh- h SLICE
e204 SG II- II- I PICTURE B fields 2 TFF time_ref 7 offset 0/0 0/0
10904 SG ii- ii- i SLICE
10944 SG JJ- gg- d PICTURE P fields 2 TFF time_ref 11 offset 0/0 0/0
13104 SG jj- gg- d SLICE
13144 SG KK- KK- K PICTURE B fields 2 TFF time_ref 9 offset 0/0 0/0
15704 SG kk- kk- k SLICE
15744 SG LL- LL- L PICTURE B fields 2 TFF time_ref 10 offset 0/0 0/0
17e44 SG ll- ll- l SLICE
17e84 SG MM- jj- g PICTURE P fields 2 TFF time_ref 14 offset 0/0 0/0
1a544 SG mm- jj- g SLICE
1a584 SG NN- NN- N PICTURE B fields 2 TFF time_ref 12 offset 0/0 0/0
1cc84 SG nn- nn- n SLICE
1ccc4 SG OO- OO- O PICTURE B fields 2 TFF time_ref 13 offset 0/0 0/0
1f384 SG oo- oo- o SLICE

```

Dans ces logs, les images sont toutes top first field (TFF)

5. -10.: Avec votre propre code et dans le langage de votre choix, implémentez un convertisseur d'images vers un format plus humainement lisi-

ble (ppm RGB est assez universel). Cela vous servira à:

- Comprendre et implémenter une conversion YUV \rightarrow RGB
- Faire une application que vous allez progressivement enrichir

Voir `ppm2pgm.py`

usage: `tvid.py [-h] --input INPUT [--fps FPS] [--ppm PPM] [--progressive]`

App to live visual mpeg flow

optional arguments:

<code>-h, --help</code>	show this help message and exit
<code>--input INPUT</code>	Folder of mpeg2dec output pgm
<code>--fps FPS</code>	Output fps
<code>--ppm PPM</code>	Output folder, save in ppm (if not show on screen)
<code>--progressive</code>	Process images as progressive

1.3 B. Jouer un flux vidéo de chaîne d'infos américaine assez notoire

1. Dans le dossier `videos/ts`, avec `ffplay`, jouez le fichier `cnn.ts`. Quel est le PID du flux vidéo?

D'après le header du fichier, le Packet IDentifier du flux vidéo est 5154.

2. `mpeg2dec` sait démultiplexer un PID de TS (option `-t`) pour tenter de le décoder comme un flux vidéo élémentaire. Connaissant le PID vidéo de `cnn.ts`, convertissez-le en pile de `pgm` via `mpeg2dec`.

Dans un dossier à la racine appelé `cnn_pgm`:

```
../tools/mpeg2dec/src/mpeg2dec -o pgm -t 5154 ../videos/ts/cnn.ts
```

3. Avec votre application, désentrelacez les `pgm` et jouez-les en cadence. Qu'observez-vous?

```
python3 tvid.py --input cnn_pgm
```

- `cnn_pgm` est le dossier qui contient la sortie de la question 2.

On peut observer que le stream est entrelacé.

4. Visualisez les flags `progressive_frame`, `top_field_first` provenant de `mpeg2dec`. A votre avis, que se passe-t-il ?

Prenons par exemple la 8ème image:

```
2c9be PICTURE P PROG fields 2 TFF pts ee43d690 dts ee43b35f time_ref 8 offset 0/0
```

`mpeg2dec` nous indique que l'image courante est progressive (PROG) et qu'elle est `top_field_first` (TFF). Nous ne pouvons ainsi pas déterminer à l'aide de `mpeg2dec` la nature de l'image courante.

Regardons alors en détails la 8ème frame.



On remarque que le texte glissant (qui n'est pas statique) montre clairement un effet d'entrelacement (ce qui n'est pas visible sur du texte statique).

Le stream décodé est un stream TNT venant d'une chaîne américaine. Or nous savons qu'un stream TNT est toujours entrelacé lors de la reception. Ensuite, ce stream a été encodé pour être sauvegardé dans un container MPEG. C'est lors de cet encodage que les images ont été encodé soit en TFF soit en PROG. Si le stream n'a pas été desentrelacé avant l'encodage, toutes les images sont donc entrelacé TFF.

De plus, visuellement toutes les images semblent entrelacées.

5. Dans `mpeg2dec`, trouvez et loggez le flag `progressive_sequence`. Si il est `== 1`, ce flag dit que dans une séquence MPEG-2, toutes les images sont progressives. Que constatez-vous? Quelle (triste) erreur de compréhension a effectué l'encodeur?

On constate qu'aucune séquence est progressive. L'erreur de compréhension est que la séquence n'est pas progressive alors que toutes les frame sont progressives.

1.4 C. Jouer un flux vidéo de chaînes de divertissement asiatiques:

1. Avec `ffplay`, trouvez les numéros de ces programmes. Relevez le troisième PID vidéo.

```
Input #0, mpegts, from '../videos/ts/ctv.ts':
```

```
Duration: 00:02:28.86, start: 43417.464456, bitrate: 9928 kb/s
```

Program 100

Metadata:

service_name :

service_provider: CTV

Stream #0:5[0x3e9]: Video: mpeg2video (Main), 1 reference frame ([2][0][0][0]

Stream #0:6[0x3ea]: Audio: mp2 ([4][0][0][0] / 0x0004), 48000 Hz, stereo, s16p

Stream #0:4[0x3eb]: Audio: mp2 ([4][0][0][0] / 0x0004), 48000 Hz, mono, s16p,

Program 101

Metadata:

service_name :

service_provider: CTV

Stream #0:2[0x3f3]: Video: mpeg2video (Main), 1 reference frame ([2][0][0][0]

Stream #0:3[0x3f4]: Audio: mp2 ([4][0][0][0] / 0x0004), 48000 Hz, stereo, s16p

Program 102

Metadata:

service_name :

service_provider: CTV

Stream #0:0[0x3fd]: Video: mpeg2video (Main), 1 reference frame ([2][0][0][0]

Stream #0:1[0x3fe]: Audio: mp2 ([4][0][0][0] / 0x0004), 48000 Hz, stereo, s16p

Program 150

Metadata:

service_name : Gemstar

service_provider: CTV

Stream #0:7[0x5dd]: Unknown: none ([13][0][0][0] / 0x000D)

Stream #0:8[0x5de]: Unknown: none ([13][0][0][0] / 0x000D)

A partir du header de ffmpeg obtenue grâce à `ffmpeg -v verbose ctv.ts`, on remarque qu'il y a plusieurs PID video pour ce stream: * 0x3e9 * 0x3f3 * 0x3fd

Celui qui nous intéresse ici est donc le troisième PID, c'est-à-dire 0x3fd.

2. Démultiplexez-le avec mpeg2dec et jouez-le via votre application.

- Démultiplexage: dans un dossier à la racine appelé `output_pgm`:
`../tools/mpeg2dec/src/mpeg2dec -o pgm -t 0x3fd ../videos/ts/ctv.ts`
- Jouez le avec notre application: `python3 tvid.py --input output_pgm --progressive`

3. Sans tenir compte du jeu d'acteurs, est-ce que le gâteau est vraiment appétissant? Autrement dit, quel problème observe t'on uniquement pendant les effets spéciaux? Pour vous aider, comparez avec vlc comme

précédemment. Selon-vous, que s'est-il passé au montage dans cette séquence précise?

Nous jouons la vidéo en progressive et nous remarquons que les effets speciaux sont entrelacés. VLC les désentrelacent correctement.

4. Avec `ffplay`, trouvez le premier PID vidéo.

Le premier PID est `0x3e9` (voir question 1)

5. Idem question C.2.

- Démultiplexage: dans un dossier à la racine appelé `output_pgm`:
`../tools/mpeg2dec/src/mpeg2dec -o pgm -t 0x3e9 ../videos/ts/ctv.ts`
- Jouez le avec notre application: `python3 tvid.py --input output_pgm --progressive`

6. En vous efforçant davantage à ignorer le jeu d'acteurs, quelle particularité rencontrez-vous avec la signalisation des images de ce flux?

```
26b2d86 SEQUENCE_REPEATED MPEG2 MP@ML 704x480 chroma 352x240 fps 29.97 maxBps 187
26b2d8e GOP DROP 7:57: 4:10
26b2d9f PICTURE I PROG fields 2 TFF pts e9a14e9b dts e9a11faf time_ref 2 offset 0
26bd79c SLICE
26bd7ae PICTURE B PROG fields 2 pts e9a13148 dts e9a13148 time_ref 0 offset 0/0 0
26becc0 SLICE
26becd2 PICTURE B PROG fields 3 RFF pts e9a13d03 dts e9a13d03 time_ref 1 offset 0
26bfef4 SLICE
26bff06 PICTURE P PROG fields 3 RFF pts e9a177aa dts e9a14e9b time_ref 5 offset 0
26c4370 SLICE
26c4382 PICTURE B PROG fields 3 TFF RFF pts e9a15a56 dts e9a15a56 time_ref 3 offs
26c579c SLICE
26c57ae PICTURE B PROG fields 2 pts e9a16bef dts e9a16bef time_ref 4 offset 0/0 0
26c6bb0 SLICE
26c6bc2 PICTURE P PROG fields 2 pts e9a1a696 dts e9a177aa time_ref 8 offset 0/0 0
26cb38c SLICE
26cb39e PICTURE B PROG fields 2 TFF pts e9a18942 dts e9a18942 time_ref 6 offset 0
26cc868 SLICE
26cc87a PICTURE B PROG fields 3 TFF RFF pts e9a194fd dts e9a194fd time_ref 7 offs
26cdfcc SLICE
26cdfde PICTURE P PROG fields 3 TFF RFF pts e9a1cfa4 dts e9a1a696 time_ref 11 off
26d2298 SLICE
26d22aa PICTURE B PROG fields 3 RFF pts e9a1b251 dts e9a1b251 time_ref 9 offset 0
```

26d3f2c SLICE
 26d3f3e PICTURE B PROG fields 2 TFF pts e9a1c3e9 dts e9a1c3e9 time_ref 10 offset
 26d6580 SLICE
 26d6592 PICTURE P PROG fields 2 TFF pts e9a1fe90 dts e9a1cfa4 time_ref 14 offset
 26daa6c SLICE
 26daa7e PICTURE B PROG fields 2 pts e9a1e13d dts e9a1e13d time_ref 12 offset 0/0
 26dc3a8 SLICE
 26dc3ba PICTURE B PROG fields 3 RFF pts e9a1ecf8 dts e9a1ecf8 time_ref 13 offset
 26dfc14 SLICE
 26dfc26 PICTURE P PROG fields 3 RFF pts e9a2279f dts e9a1fe90 time_ref 17 offset
 26e4120 SLICE
 26e4132 PICTURE B PROG fields 3 TFF RFF pts e9a20a4b dts e9a20a4b time_ref 15 off
 26e59ec SLICE
 26e59fe PICTURE B PROG fields 2 pts e9a21be4 dts e9a21be4 time_ref 16 offset 0/0
 26e7fa8 SLICE
 26e7fba PICTURE P PROG fields 2 pts e9a2568b dts e9a2279f time_ref 20 offset 0/0
 26ec5cc SLICE
 26ec5de PICTURE B PROG fields 2 TFF pts e9a23937 dts e9a23937 time_ref 18 offset
 26edef8 SLICE
 26edf0a PICTURE B PROG fields 3 TFF RFF pts e9a244f2 dts e9a244f2 time_ref 19 off
 26f175c SLICE
 26f176e PICTURE P PROG fields 2 TFF pts e9a27f99 dts e9a2568b time_ref 23 offset
 26f6118 SLICE
 26f612a PICTURE B PROG fields 3 RFF pts e9a26246 dts e9a26246 time_ref 21 offset
 26f8b24 SLICE
 26f8b36 PICTURE B PROG fields 2 TFF pts e9a273de dts e9a273de time_ref 22 offset

On peut remarquer que certaines images sont à la fois Progressive (PROG),
 Top First Field (TFF) et Repeat First Field (RFF).