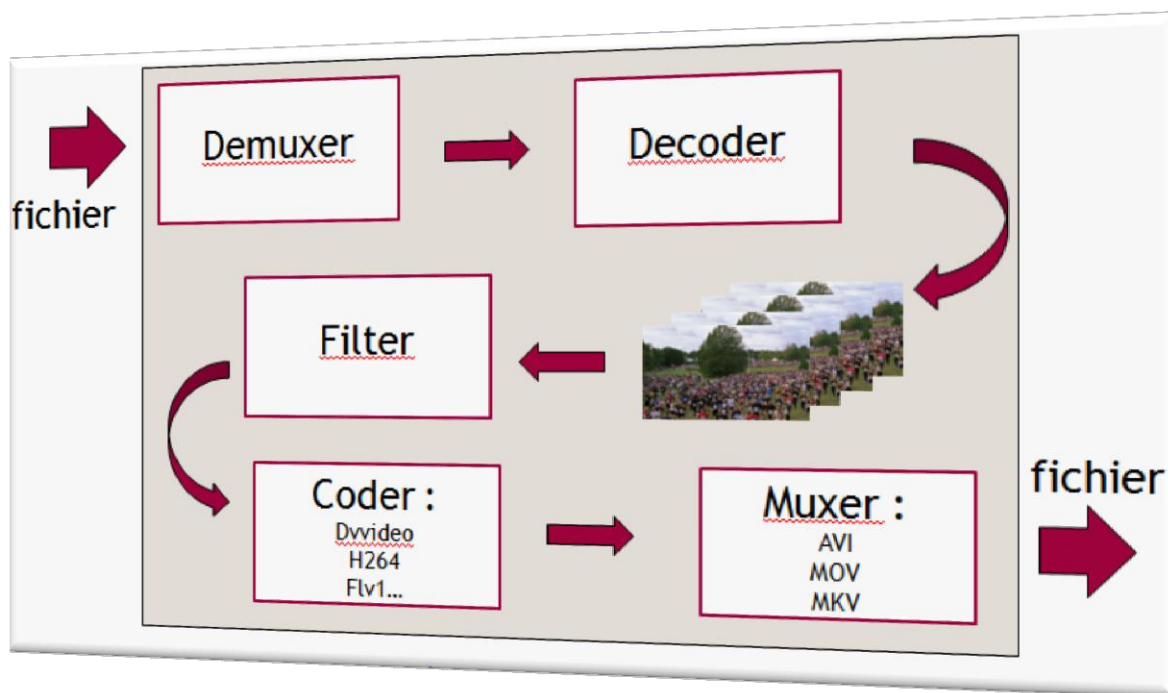




```

Administrator: C:\Windows\system32\cmd.exe

C:\>ffmpeg -i E:\EBU_5seq_MonCompress_576i.avi
ffmpeg version N-64919-ga613257 Copyright (c) 2000-2014 the FFmpeg developers
built on Jul 23 2014 00:35:22 with gcc 4.8.3 (GCC)
configuration: --enable-gpl --enable-version3 --disable-w32threads --enable-av
--enable-libfreetype --enable-libluray --enable-frei0r --enable-gnutls --enab
--enable-libmp3lame --enable-libgsm --enable-libb2b --enable-libcaca --
--enable-libopenjpeg --enable-libopus --enable-librtmp --enable-libschroedinge
--enable-libsoxr --enable-libspeex --enable-libtheora --enable-libtwolame --en
--enable-libvpx --enable-libvaapi --enable-libwebp --enable-libx264 --enable-
libx265 --enable-libxavs --enable-libxvid --enable-decklink --enable-zlib
libavutil 52. 92.101 / 52. 92.101
libavcodec 55. 69.100 / 55. 69.100
libavformat 55. 48.101 / 55. 48.101
libavdevice 55. 13.102 / 55. 13.102
libavfilter 4. 11.102 / 4. 11.102
libswscale 2. 6.100 / 2. 6.100
libswresample 0. 19.100 / 0. 19.100
libpostproc 52. 3.100 / 52. 3.100
Input #0, avi, from 'E:\EBU_5seq_MonCompress_576i.avi':
Duration: 00:00:50.00, start: 0.000000, bitrate: 165894 kb/s
Stream #0:0: Video: rawvideo (UYVY / 0x32595559), yuyv422, 720x576, 166020 k
t least one output file must be specified
  
```



Christophe NELSON : Version 3.1

1. Objectifs :

L'objectif de ce module est de comprendre les avantages des outils vidéo en ligne de commande.

Même si la vidéo est par essence visuelle, ces outils peuvent être très pratiques pour comprendre, normaliser et automatiser des actions de codage.

2. Outils nécessaires :

2.1. logiciels

- AVI synth (<http://avisynth.org/>)
- VirtualDUB (<http://www.virtualdub.org>)
- AVSP (<http://avisynth.org/qwerpoi/>)
- FFMPEG (<http://ffmpeg.org/>)
- FFMBC (<http://code.google.com/p/ffmbc/>)
- FFdshow (<http://ffdshow-tryout.sourceforge.net/>)
- Matroska splitter (<http://haali.su/mkv/>)

2.2. Système d'exploitation

Nous allons essentiellement travailler sous **Windows** (XP, vista ou seven)

AVIsynth ne fonctionne dans sa version actuelle que sous Windows. Un projet de version 3 devrait évoluer vers un multiplateforme mais ce projet semble ralenti. Il existe également un projet avxsynth, un portage de avisynth sous linux.

<https://github.com/avxsynth/avxsynth/wiki>

FFmpeg est multiplateforme

La partie automatisé est, elle aussi, prévue pour **Windows** mais est facilement transposable sous linux

3. Introduction aux lignes de commandes

- Inconvénients :
 - > C'est laid
- Avantages :
 - > C'est plus facile de reproduire les actions
 - > On comprend mieux ce qu'on fait
 - > Certaines fonctions ne sont disponibles qu'en ligne de commande

- > C'est la frime ;-)

Utiliser des lignes de commandes peut paraître austère au premier abord, surtout dans un domaine aussi visuel que la vidéo ou la télévision. Cependant, même si le montage et les effets ne peuvent se faire qu'avec des outils visuels, on peut trouver un ensemble d'opérations répétitives qui vont pouvoir être automatisées grâce à ces outils en ligne de commande.

- Remarque

Interface en lignes de commande est souvent appelée CLI (Command Line Interface) par opposition à l'interface graphique GUI (graphical user interface).

3.1. Pourquoi utiliser des lignes de commande

- Autres avantages :

- > Certains codeurs GUI (Graphic User Interface) ne savent pas :
- > Ouvrir certains formats de fichier (ex : MKV, MTS...)
- > Décoder certaines sources vidéo (ex AVCHD, snow, dirac, webM)
- > Appliquer des filtres (ex retailage, désentrelacement)



Différentes fonctions accessibles en CLI

- Remarque

- > La plupart de ces outils sont "Open Source", c'est à dire que le code est ouvert, donc ces outils évoluent très vite, grâce à une communauté d'utilisateurs et souvent poussés par les professionnels du domaine.
- > De plus ils sont gratuits, ce qui n'est pas négligeable.

4. Encodeur en lignes de commande

Il existe de nombreux encodeurs en ligne de commande

x264 est issu du projet VideoLAN permettant de coder des flux vidéo H.264. C'est un codec uniquement vidéo

ffmpeg est un codeur plus complet qui gère de nombreux formats. Il est très utilisé et certains grands groupes audiovisuels participent à son développement.

Mencoder est inclus dans le pack Mplayer.

Nous avons décidé d'utiliser ffmpeg, car il est très utilisé et en constant développement. Il est devenu une référence.

4.1. Présentation de FFMPEG

FFmpeg est une collection de logiciels libres dédiés au traitement de flux audio et/ou vidéo (enregistrement, lecture ou conversion d'un format à un autre).

Cette collection est constituée de logiciels :

FFmpeg : Outil en ligne de commande pour convertir des vidéos mais il peut aussi capturer des flux en temps réel

FFplay : Un simple lecteur vidéo

FFserver : Un serveur de streaming en HTTP et RTSP

FFprobe : Un analyseur vidéo

Cette collection contient également des bibliothèques. Ces bibliothèques sont utilisées par de nombreux logiciels.

Par exemple VLC s'appuie sur ces bibliothèques pour décoder et afficher de nombreuses vidéos.

Libavcodec : bibliothèque contenant de nombreux codeurs et décodeurs vidéo. Elle contient plus de 100 codecs !

libavformat : bibliothèque contenant des muxers, démuxers, c'est à dire l'outil permettant d'ouvrir de nombreux types de fichiers audio/vidéo

libswscale : bibliothèque contenant des routines de traitement de l'image et du son

FFmpeg est multiplateforme. Il existe pour linux pour windows et pour mac OS

- Attention

Ne pas confondre FFmpeg et FFmpegX qui est une interface graphique pour FFmpeg.

4.1.1. Compilation

Le site officiel de FFmpeg fournit seulement le code source. On peut accéder à la toute dernière version de FFmpeg terminée dans la nuit mais il n'y a pas de version compilée ! On peut télécharger les sources et donc compiler FFmpeg avec les bibliothèques de son

choix, mais ce n'est pas forcément simple et on rentre plus dans le domaine de la programmation.

<https://www ffmpeg.org/download.html>

Il est cependant possible de récupérer des versions compilées, récentes de FFmpeg sur certains sites.

Par exemple :

<http://ffmpeg.zeranoe.com/builds/>

Il y a 3 versions disponibles :

- > **Static** Contient les exécutables (ffmpeg, ffprobe, ffplay).
- > **Shared** Contient des exécutables avec les bibliothèques séparées pour permettre l'utilisation dans d'autres programmes.
- > **Dev** Contient des exécutables avec les bibliothèques séparées pour permettre le développement d'autres programmes.

Le plus simple est d'utiliser STATIC

4.1.2. Installation

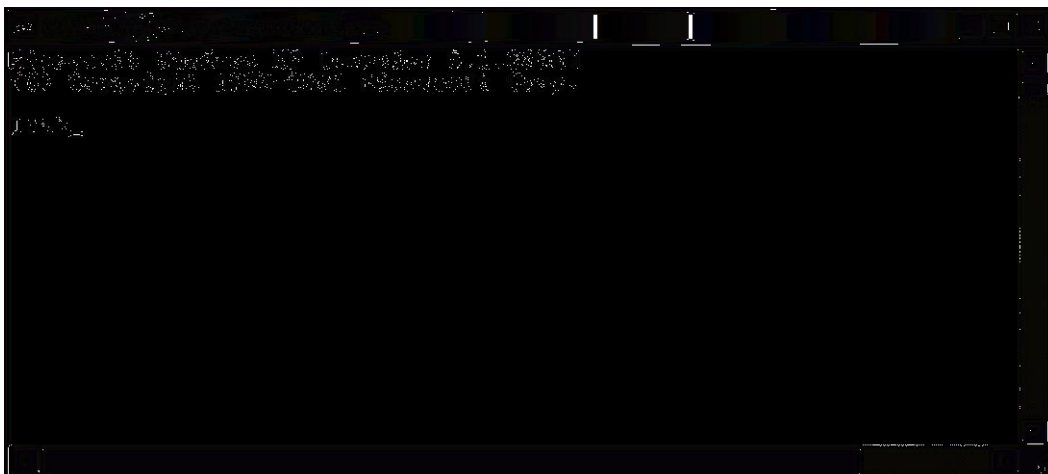
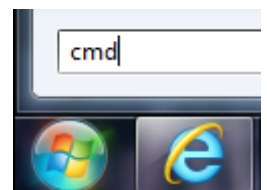
FFMPEG est un exécutable. Pour le lancer par une ligne de commande, il faut normalement indiquer son chemin d'accès. Ça peut être long et fastidieux de le faire à chaque fois.

- > Copier coller FFmpeg dans le dossier C:\Windows\System32
- > FFmpeg est maintenant accessible sans indiquer de chemin.

4.1.3. L'interface en ligne de commande

Pour utiliser ffmpeg, il faut d'abord ouvrir une invite de commande : Pour cela, cliquer sur le bouton « fenêtre » puis taper cmd (pour command) dans la ligne de recherche.

La fenêtre de commande s'ouvre :



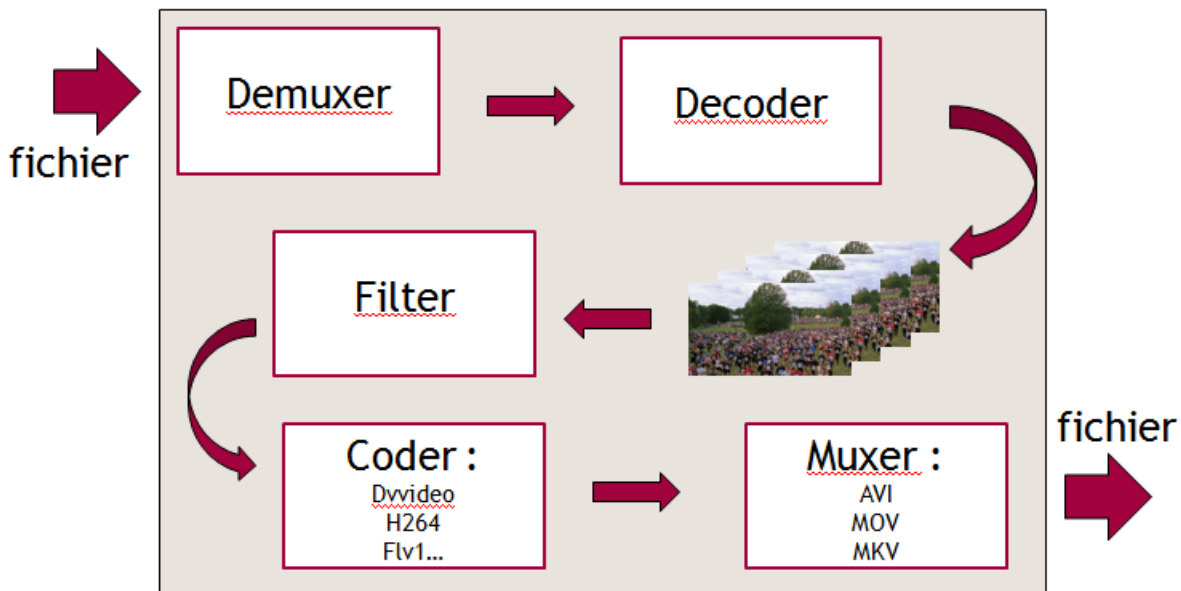
Exemple de Command Line Interface

Vous pouvez maintenant naviguer sur les disques :
K: pour aller sur le disque "Stagiaire"
dir pour lister les fichiers et dossiers où nous sommes
cd <NomDossier> permet de rentrer dans le dossier

4.1.4. Structure de FFMPEG

FFmpeg intègre tous les éléments nécessaires de la chaîne de transcodage :

- > Ouverture de nombreux fichiers (Demuxer)
- > Décodage des images (decoder)
- > Traitements (filter)
- > Encodage vidéo (Coder)
- > Encapsulation dans de nombreux formats de fichier (muxer)



4.1.5. La documentation de FFmpeg

La documentation de FFmpeg est disponible sur le site de FFmpeg mais pour être sûr d'avoir les options correspondant à la version utilisée la commande est :

```
FFmpeg.exe -h
```

Il est possible d'envoyer ce contenu dans un fichier texte :

```
FFmpeg.exe -h >> FFmpeg_aide.txt
```

Liste les formats de fichiers reconnus par le demuxer et le muxer de FFmpeg :

```
FFmpeg.exe -formats >> FFmpeg_formats.txt
```

Liste des codecs reconnus par le codeur et le décodeur de FFmpeg :

```
FFmpeg.exe -codecs >> FFmpeg_codecs.txt
```

Enfin, l'aide peut également être spécifique à un codeur, décodeur, muxeur ou démuxeur :

```
FFmpeg.exe -h demuxer=<Nom du démuxeur>
```

```
FFmpeg.exe -h decoder=<Nom du décodeur>
```

```
FFmpeg.exe -h encoder=<Nom du codeur>
```

```
FFmpeg.exe -h muxer=<Nom du muxeur>
```

Par exemple :

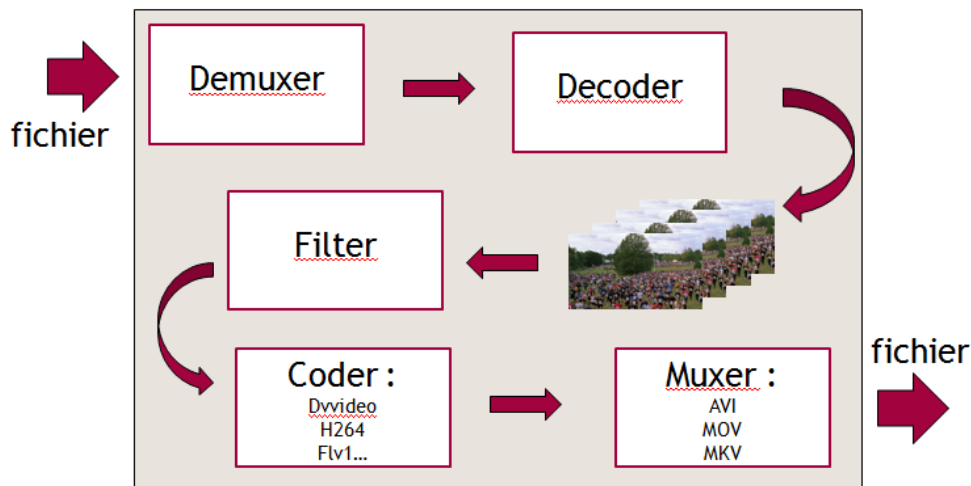
```
FFmpeg.exe -h muxer=webm
```

4.2. Syntaxe de FFMPEG (Coder et Muxer)

```
ffmpeg.exe -i <Fichier_source> -filtres <Fichier_de_sortie>
```

- i permet de spécifier un fichier source (demuxer & decoder)
- options seront des paramètres de modification de la source
- <Fichier_de_sortie> permet d'indiquer le fichier de sortie (coder & muxer)

Note : en fonction de l'extension du fichier FFmpeg choisi l'encodage associé



4.2.1. Exemple : Obtenir des informations sur la vidéo

Pour obtenir des informations sur un fichier video, il suffit de ne pas mettre de fichier de sortie

```
ffmpeg.exe -i <Fichier_source>
```

- i <Fichier_source> : ouvre (-i pour Input) la vidéo
- <Fichier_source> peut être un fichier vidéo. Ce nom doit comporter le chemin d'accès ou le chemin relatif

Exemple :

Récupérez les informations par ffmpeg sur le fichier E:\EBU_5seq_NonCompress_576i.avi

```

C:\Windows\system32\cmd.exe
E:\>ffmpeg -i E:\EBU_5seq_NonCompress_576i.avi
ffmpeg version N-64919-ga613257 Copyright (c) 2000-2014 the FFmpeg developers
  built on Jul 23 2014 00:35:22 with gcc 4.8.3 (GCC)
  configuration: --enable-gpl --enable-version3 --disable-w32threads --enable-av
  isynth --enable-bzlib --enable-fontconfig --enable-frei0r --enable-gnutls --enab
  le-iconv --enable-libass --enable-libbluray --enable-libs2b --enable-libcaca --
  enable-libfreetype --enable-libgme --enable-libgsm --enable-libilbc --enable-lib
  modplug --enable-libmp3lame --enable-libopencore-amrnb --enable-libopencore-amrw
  b --enable-libopenjpeg --enable-libopus --enable-librtmp --enable-libschröedinge
  r --enable-libsoxr --enable-libspeex --enable-libtheora --enable-libtwolame --en
  able-libvidstab --enable-libvo-aacenc --enable-libvo-amrwbenc --enable-libvorbis
  --enable-libvpx --enable-libwaupack --enable-libwebp --enable-libx264 --enable-
  libx265 --enable-libxavs --enable-libxvid --enable-decklink --enable-zlib
  libavutil      52. 92.101 / 52. 92.101
  libavcodec     55. 69.100 / 55. 69.100
  libavformat    55. 48.101 / 55. 48.101
  libavdevice    55. 13.102 / 55. 13.102
  libavfilter     4. 11.102 / 4. 11.102
  libswscale     2.  6.100 / 2.  6.100
  libswresample  0. 19.100 / 0. 19.100
  libpostproc    52.  3.100 / 52.  3.100
Input #0, avi, from 'E:\EBU_5seq_NonCompress_576i.avi':
  Duration: 00:00:50.00, start: 0.000000, bitrate: 165894 kb/s
  Stream #0:0: Video: rawvideo (YUY2 / 0x32595559), yuyv422, 720x576, 166020 k
  b/s, 25 fps, 25 tbr, 25 tbn, 25 tbc
At least one output file must be specified
E:\>

```

Quelle est la durée du fichier source :

Quel est le débit :

la définition de l'image :

le frame rate :

4.2.2. Exemple : Convertir en WebM

Dans les versions récentes de FFmpeg, il est possible d'encoder dans le format libre de Google : le WebM

Copier le fichier 10s_HD_Non_Compressée.avi sur k:

```
ffmpeg.exe -i k:\10s_HD_Non_Compressée.avi output.webm
```

Ne pas fermer la fenêtre de commande !

Ouvrir le fichier output.webm dans VLC

Quel est sa qualité ?

Quel est le débit demandé ? (si besoin remonter dans la fenêtre de commande pour voir les paramètres d'encodage.

Quel est le codec choisi ?


```
Output #0, webm, to 'output.webm':
  Metadata:
    encoder      : Lavf55.48.101
  Stream #0:0: Video: vp8 (libvpx), yuv420p, 1920x1080, q=-1--1, 200 kb/s, 25
  fps, 1k tbn, 25 tbc
  Metadata:
    encoder      : Lavc55.69.100 libvpx
  Stream mapping:
    Stream #0:0 -> #0:0 (rawvideo (native) -> vp8 (libvpx))
```

4.2.3. Réglage du débit

Dans une commande, FFmpeg est suivi d'options. Pour choisir le débit on utilise -b pour bitrate.

- > -b : configuration du débit binaire (par défaut 200 kbps)
- > -b:v : configuration du débit video
- > -b:a : configuration du débit audio

le débit peut être exprimé en

bit/s : pas d'unité (défaut)

Kb/s : K

Mb/s : M

Gb/s : G

Choisir une des commandes ci dessous pour encoder le fichier à 7.5 Mb/s

```
ffmpeg.exe -i k:\10s_HD_Non_Compressee.avi -b:v 7500000 output.webm
```

```
ffmpeg.exe -i k:\10s_HD_Non_Compressee.avi -b:v 7500K output.webm
```

```
ffmpeg.exe -i k:\10s_HD_Non_Compressee.avi -b:v 7.5M output.webm
```

```
ffmpeg.exe -i k:\10s_HD_Non_Compressee.avi -b:v 0.0075G output.webm
```

Vérifier que la qualité est meilleure

4.2.4. Choix du Codec (Convertir en format DV)

On peut utiliser des préréglages existants pour des formats normalisés (VCD, SVCD, DVD, DV et DV50) pour lesquels tous les paramètres de fréquence image, de débit, d'aspect ratio sont déjà inclus :

- > vcd, pal-vcd, ntsc-vcd, film-vcd
- > svcd, pal-svcd, ntsc-svcd, film-svcd
- > dvd, pal-dvd, ntsc-dvd, film-dvd
- > dv, pal-dv, ntsc-dv, film-dv
- > dv50, pal-dv50, ntsc-dv50, film-dv50

On utilise alors -target

Voici une ligne de commande pour convertir une vidéo en DV pal :

```
ffmpeg.exe -i k:\10s_HD_Non_Compressee.avi -target pal-dv video_DV.avi
```

- > -i input.avs : ouvre (-i pour Input) la vidéo
- > -target pal-dv : cible une préconfiguration du format de sortie en « pal-dv »
- > video_DV.avi : nom et conteneur de la vidéo transformée

4.2.5. Choix du Codec (Encoder en h264)

Pour choisir le codec on utilise -c

- > -c:v choix du codec video
- > -c:a choix du codec audio

Voir la liste des codecs dans le chapitre 4.1.5

Coder la video en H264

```
ffmpeg.exe -i k:\10s_HD_Non_Compressee.avi -c:v h264 video_H264.mp4
```

- > -i <Fichier_source> : ouvre (-i pour Input) la vidéo
- > -c:v h264 : choisit le codec h264

Pour accéder à plus de paramètres, il faut utiliser le codec libre x264 directement intégré dans FFmpeg sous forme de librairie : libx264

```
ffmpeg.exe -i k:\10s_HD_Non_Compressee.avi -c:v libx264 -preset slow -x264-params qp=22 video_H264.mp4
```

- > -i <Fichier_source> : ouvre (-i pour Input) la vidéo
- > -c:v Libx264 : On va utiliser la librairie libx264 pour le codage (-c:) de la video (v)
- > -preset:v slow : on utilise des préréglages lents. Ils permettent de prendre le temps de bien encoder les différents preset sont ultrafast, superfast, veryfast, faster, fast, medium, slow, slower, veryslow, placebo.
- > -x264-params qp=22 définit le niveau de quantification. On ne choisit plus le débit de sortie mais la qualité des images. Le débit dépendra donc du contenu de la vidéo

Quel est le débit du fichier obtenu ?

■ Note :

Le « placebo » permet d'améliorer la qualité de 1% par rapport au « veryslow »

Le « veryslow » permet d'améliorer la qualité de 3% par rapport au « slower »

Le « slower » permet d'améliorer la qualité de 5% par rapport au « slow »

Le « slow » permet d'améliorer la qualité de 5% à 10% par rapport au « medium »

Exercice : Ecrire une ligne de commande pour faire un flux H264 à 7.5 Mb/s de la meilleure qualité possible

4.2.6. Choix du Codec (Video non compressé)

Comme ffmpeg décode de nombreux codecs, il peut être utile de l'utiliser pour décoder une vidéo et l'enregistrer dans un format non compressé :

```
ffmpeg.exe -i k:\10s_HD_Non_Compressée.avi -c:v rawvideo  
video_Non_Compressée.avi
```

4.2.7. Forcer l'écriture

Une fonction simple mais utile est de pouvoir éviter la question "overwrite ? [y/N]" surtout quand on travaille régulièrement sur le même fichier de sortie. Pour cela il suffit de rajouter -y (pour yes) dans la ligne de commande.

```
ffmpeg.exe -i k:\10s_HD_Non_Compressée.avi -c:v rawvideo -y  
video_Non_Compressée.avi
```

4.2.8. Déwrapper-Wrapper

Nous pouvons utiliser FFmpeg pour faire du démultiplexage, remultiplexage sans recompression :

```
ffmpeg -i video_H264.mp4 -c:v copy -c:a copy -vbsf h264_mp4toannexb -y  
video_H264.ts
```

-c:v copy force le codec vidéo en "copie"
-c:a copy force le codec audio en "copie"
-vbsf h264_mp4toannexb permet de filtrer les flux de type H.264 pour qu'ils soient correctement multiplexés.

4.2.9. Exercice Encoder en H265

Objectif : Convertir la séquence 10s_HD_Non_Compressée.avi en une vidéo compressée en HEVC/H265 avec une quantification de 22 dans un fichier .mp4.

Le X265 a été intégré à FFmpeg sous forme de libx265.

Utiliser un preset slow.

Noter la ligne de commande :

Regarder le temps d'encodage.
Comparer avec le fichier obtenu précédemment en H264.
Commenter les résultats :

4.2.10. Extraire des images d'une vidéo

```
ffmpeg -i video.mpg -sameq -f image2 "test-%4d.jpg"
```

- > -i : indique le fichier de départ (input file), le nom du fichier suit
- > -sameq permet de garder la même qualité que la source
- > -f : forcer le format de sortie (image2 signifie séquence d'image)
- > %04d signifie que les fichiers seront enregistrés avec 4 chiffres croissants (sous la forme test-0001.jpg, test-0002.jpg, test-0003.jpg, etc.)

4.3. Syntaxe de FFMPEG (Filter)

Outre les paramètres d'encodage et de décodage, ffmpeg contient des filtres assez efficaces.

4.3.1. Quelques options vidéo :

- > -r : définit le nombre d'images par seconde
- > -s (LxH) : configuration de la taille de l'image
- > -aspect : configuration du format d'affichage (4:3, 16:9 ou 1.3333, 1.7777)
- > -qmin : détermine le seuil minimum du débit binaire variable (VBR)
- > -qmax : détermine le seuil maximum du débit binaire variable (VBR)
- > -vframes : définit le nombre de frames de la vidéo à traiter (pour ne prendre qu'un extrait).
- > -croptop size : espace extérieur haut (en pixels)
- > -cropbottom size : espace extérieur bas (en pixels)
- > -cropleft size : espace extérieur gauche (en pixels)
- > -cropright size : espace extérieur droit (en pixels)
- > -padtop size : marge du haut (en pixels)
- > -padbottom size : marge du bas (en pixels)
- > -padleft size : marge de gauche (en pixels)
- > -padright size : marge de droite (en pixels)
- > -padcolor color : couleur des marges (Hex 000000 thru FFFFFFFF)

Quelques options audio :

- > -ar : configuration de la fréquence d'échantillonnage (par défaut 44100 Hz)
- > -ac : configure le nombre de canaux (mono-stéréo)
- > -vol volume modifie le volume audio (défaut=256)
- > -alang code définit le langage du flux audio (3 lettres)

4.3.2. Exemple : resize

```
ffmpeg.exe -i k:\10s_HD_Non_Compressee.avi -s 1280x720 output.mp4
```

> -s largeur x hauteur

4.3.1. Video filter

Si on veut travailler avec plus de possibilités, il vaut mieux utiliser un "video filter" que l'on appelle avec -vf.

Par exemple -vf peut utiliser l'option scale : -vf scale=y:x

-vf scale

```
ffmpeg.exe -i k:\10s_HD_Non_Compressee.avi -vf scale=1280:720  
output.mp4
```

Cette ligne a le même effet que la précédente

mais on peut travailler avec des variables :

iw : largeur de la source

ih : hauteur de la source

```
ffmpeg.exe -i k:\10s_HD_Non_Compressee.avi -vf scale=iw/2:ih/2  
output.mp4
```

Divise la taille de l'image par 2

4.3.2. Encoder un nombre d'images en vidéo

```
ffmpeg.exe -r 25 -an -i Course-%04d.bmp video.mpg
```

- > -r 25 définit le nb d'images par seconde
- > -an pas de son
- > %04d signifie que les images sont numérotées dans le nom sur 4 caractères

4.3.3. Insérer un logo

On peut aller très loin avec FFmpeg. Voici par exemple comment incruster un logo.
Copier ina2.png sur K :



Enseignement
Formation
Recherche

Pour la ligne de commande, il faut 2 entrées : la vidéo, et le logo. Nous utiliserons donc 2 fois la commande -i

Nous utiliserons ensuite un filtre complexe `-filter_complex` qui utilise 2 entrées et permet avec l'option `overlay` de superposer 2 images :

```
ffmpeg -i <video_source> -i <Image_Logo> -filter_complex overlay=X:Y <fichier_de_sortie>
```

- > X: position horizontale de la seconde source
- > Y : position verticale de la seconde source

```
ffmpeg -i k:\10s_HD_Non_Compressee.avi -i k:\ina2.png -filter_complex overlay output.mp4
```

Visualiser le résultat. Par défaut les valeurs de X et Y sont à 0

Overlay peut utiliser les variables suivantes :

W largeur de la source vidéo

H hauteur de la source vidéo

w largeur du logo

h hauteur du logo

Placer le logo à droite de l'image grâce à ces variables

4.4. Pour aller plus loin

4.4.1. Usage de FFplay

Pour mettre en œuvre ces filtres, pour les tester avant de lancer l'encodage je vous conseille d'utiliser FFplay, le player vidéo de FFmpeg.

FFplay fonctionne comme FFmpeg, mais au lieu de sortir un fichier, il sort l'image directement à l'écran. C'est donc une bonne façon de visualiser le résultat de ces filtres sans perdre du temps à encoder chaque fois

```
ffplay.exe -i k:\10s_HD_Non_Compressee.avi
```

Pendant le visionnage quelques fonctions peuvent-être pratiques :



Avance de 10s



Reculé de 10s



Avance de 1min



Recul de 1min

F permet de se mettre en plein écran

A permet de basculer entre les pistes audio (s'il y en a plusieurs)

V permet de basculer entre les pistes vidéo (flux ts par exemple)

Un clic dans l'image va au temps correspondant à la position de la souris : par exemple un clic à gauche de l'image, un clic à droite de l'image va à la fin

4.4.2. FFprobe

Avec FFmpeg est fournit FFprobe qui permet de récolter des infos d'encodage d'un fichier vidéo.

```
ffprobe -show_streams EBU_50s_NonComprese.avi
```

Vous pouvez également ajouter à la fin.

```
>>info.txt
```

Mettre ces infos dans un fichier texte, plus exploitable en programmation.

4.4.3. FFserver

Sur le site de ffmpeg on peut également trouver FFserver, un encodeur de flux temps réel.

4.4.4. FFMBC

A partir des sources de FFmpeg, certains ont fait évoluer l'application vers des usages plus Broadcast. Est apparu FFMedia BroadCast : FFMBC

Avec FFmbc vous pouvez :

Créer des XDcam HD422 en .mov ou .mxf

Créer des XDcam IMX/D-10 en .mov ou .mxf

Créer des DNxHD en .mov

Créer du ProRes 422 et 4444 en .mov

Transcoder de l'AVCIntra 50 ou 100

Transcoder de l'AVCHD

Créer des QuickTime contenant des pistes de TimeCode

<http://code.google.com/p/ffmbc/>

- Exemple : créer un PAD HD en XDcamHD422

```
ffmbc -i <Fichier_Source> -target xdcamhd422 <Video_finale.mxf>
```

5. Le serveur d'images

5.1. Qu'est-ce qu'un FrameServer ?

Un serveur d'image (FrameServer en anglais) est une application qui délivre une vidéo directement à une autre application. Il se charge de créer un pseudo-fichier que les autres programmes peuvent lire comme si c'était un très gros fichier AVI (généralement non compressé).

- C'est un serveur : il va rendre disponible un flux vidéo et audio à partir d'un fichier ou autre pour un autre logiciel (le client).
- Le client recevra la vidéo image par image, à son rythme.
- Aucun fichier intermédiaire ou temporaire n'est créé pour aller d'un logiciel à l'autre.
- Le FrameServer peut également appliquer des traitements sur la vidéo avant de la transmettre au client (désentrelacement, filtrage, redimensionnement)

5.1.1. Fonctionnement d'un FrameServer :



La source (que ce soit un dvd, un fichier sur le disque dur local, distant ou un flux comme du streaming vidéo) est ouverte, décodée et éventuellement traitée par le FrameServer. Il envoie ensuite cette vidéo vers le logiciel client (encodeur, afficheur, etc.)

5.1.2. Exemple de FrameServer : AVISynth

- Beaucoup de logiciels reconnaissent ce FrameServer (virtualDub, ffmpeg, AVIDemux, Megui, MSU Video Quality measurement...)
- Il est Open-Source donc évolutif
- Il intègre de nombreux filtres très utiles
- Il est très documenté
- Il est gratuit

5.1.3. Installation d'AVIsynth

L'installateur d'AVIsynth se trouve sur sourceforge

<http://sourceforge.net/projects/avisynth2/files/>

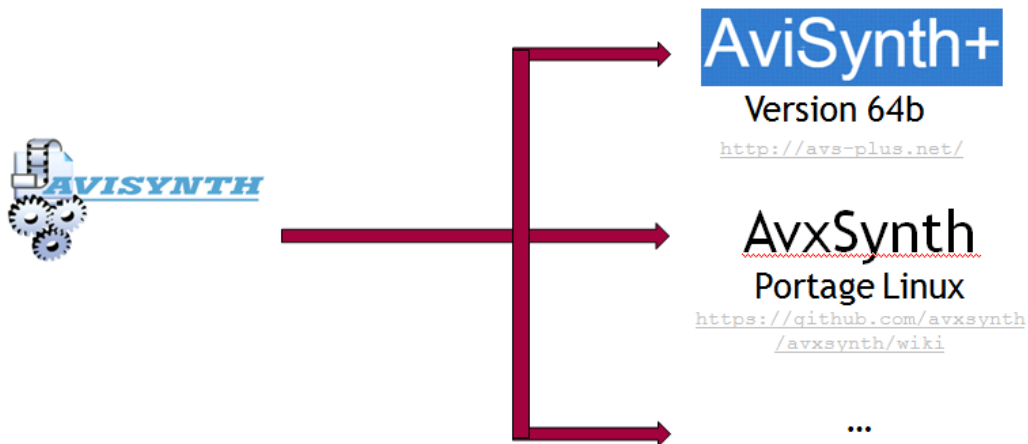
Il suffit de double-cliquer sur l'installateur et il s'installe.

AVISynth plus est une « fork », une reprise du projet non officielle, qui améliore certains points, et permet notamment de fonctionner en 64b.

<http://avs-plus.net/>

Il existe également un portage linux : AvxSynth

<https://github.com/avxsynth/avxsynth/wiki>



■ Remarque

AviSynth ne propose pas d'environnement graphique (GUI), mais se base à la place sur un système de scripts que nous allons voir maintenant.

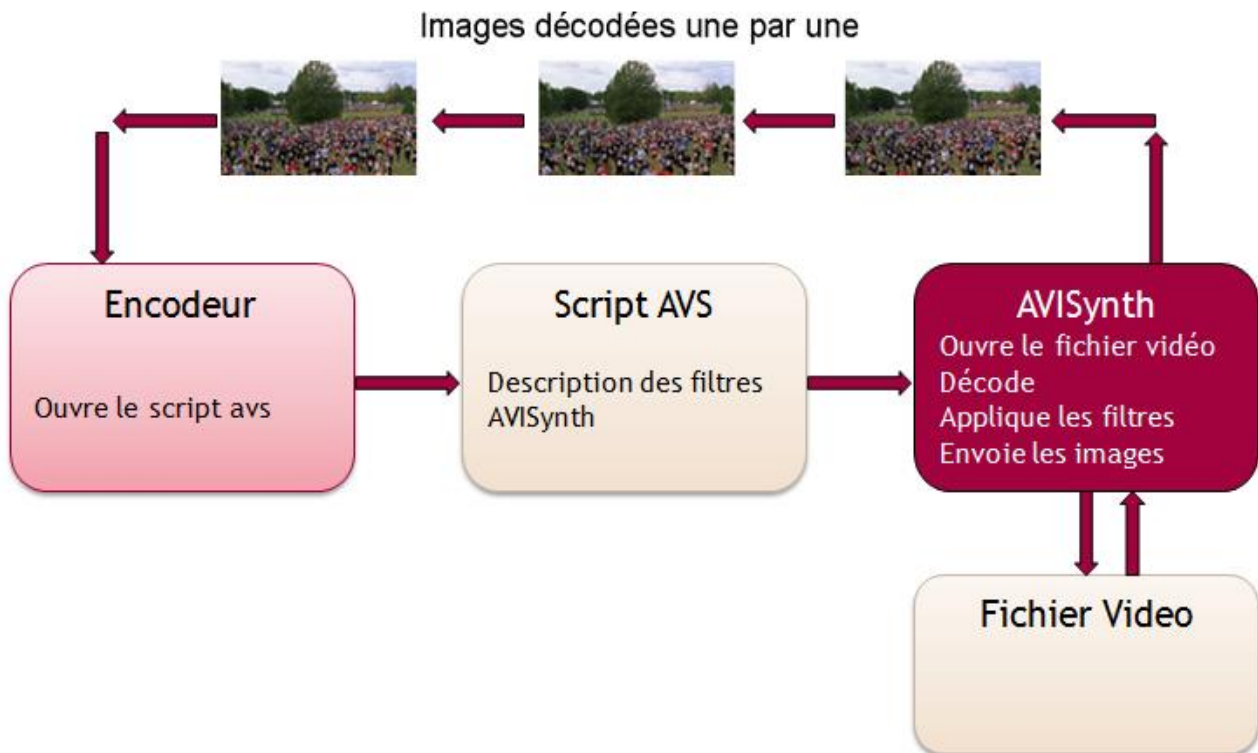
Il n'y a pas d'icône dans le menu démarrer, pas de programme affiché, c'est un peu déroutant au début.

5.2. Utilisation d'AVISynth

Dans un logiciel d'encodage par exemple, compatible avec AVIsynth, au lieu d'ouvrir une vidéo, on va ouvrir un script AVS. Ceci à pour effet de démarrer le frameserver.

Le script AVS contient des "commandes" pour notre frameserver

Le frameserver suit les instructions contenues dans le fichier AVS. Elles vont généralement lui dire d'ouvrir un fichier vidéo, puis appliquer des filtres à cette vidéo Enfin il va transmettre au logiciel de départ, ici un encodeur, les images une par une, non compressées, sans passer par un fichier intermédiaire.



- Définition : Script AVS

Le script AVS est ce qui donne des instructions à AVISynth. C'est un fichier texte avec une extension de fichier .avs

Grâce à ce script AVISynth saura quel fichier ouvrir, comment le décoder, comment le transformer avant de l'envoyer vers un autre logiciel.

5.2.1. Script AVISynth

Un script AVS est décomposé en 2 parties :

- Des commandes pour l'importation des sources (INDISPENDABLES)
- Des commandes pour le traitement de l'image (filtres : redimensionnement, découpage, filtrage)

Commande	Types de fichiers
AVISource	Ouvre un fichier avi
ImageSource	Ce filtre produit un clip vidéo à partir d'images
WavSource	Ouvre un fichier WAV ou la piste audio d'un fichier AVI.
DirectShowSource	Ouvre des fichiers en s'appuyant sur DirectShow.

Exemple : des commandes pour l'importation des sources

Les trois premiers sont des « filtres » internes d'avisynth.

Le DirectShowSource permet d'ouvrir certains types de fichier (non prévu en natif) en utilisant les filtres externes de directshow.

Quand on installe un codec dans windows, on installe en fait ce codec dans une partie appelée directshow.

Pour pouvoir ouvrir beaucoup de formats, il peut être intéressant d'ouvrir une librairie de codec, voire une librairie de conteneurs.

Je vous conseille :

FFdshow : librairie libre de codecs

<http://ffdshow-tryout.sourceforge.net/>

Matroska splitter : permet d'ouvrir Matroska, MP4, AVI, OGG/OGM, MPEG TS

<http://haali.su/mkv/>

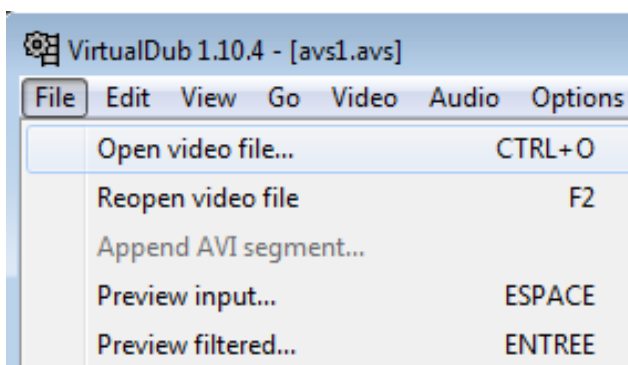
5.2.2. 1^{er} script : ouvrir une source vidéo

Nous allons créer notre premier script AVS. L'objectif sera simplement d'ouvrir une vidéo. Ce script pourra donc servir à ouvrir cette vidéo dans n'importe quel logiciel compatible AVIsynth

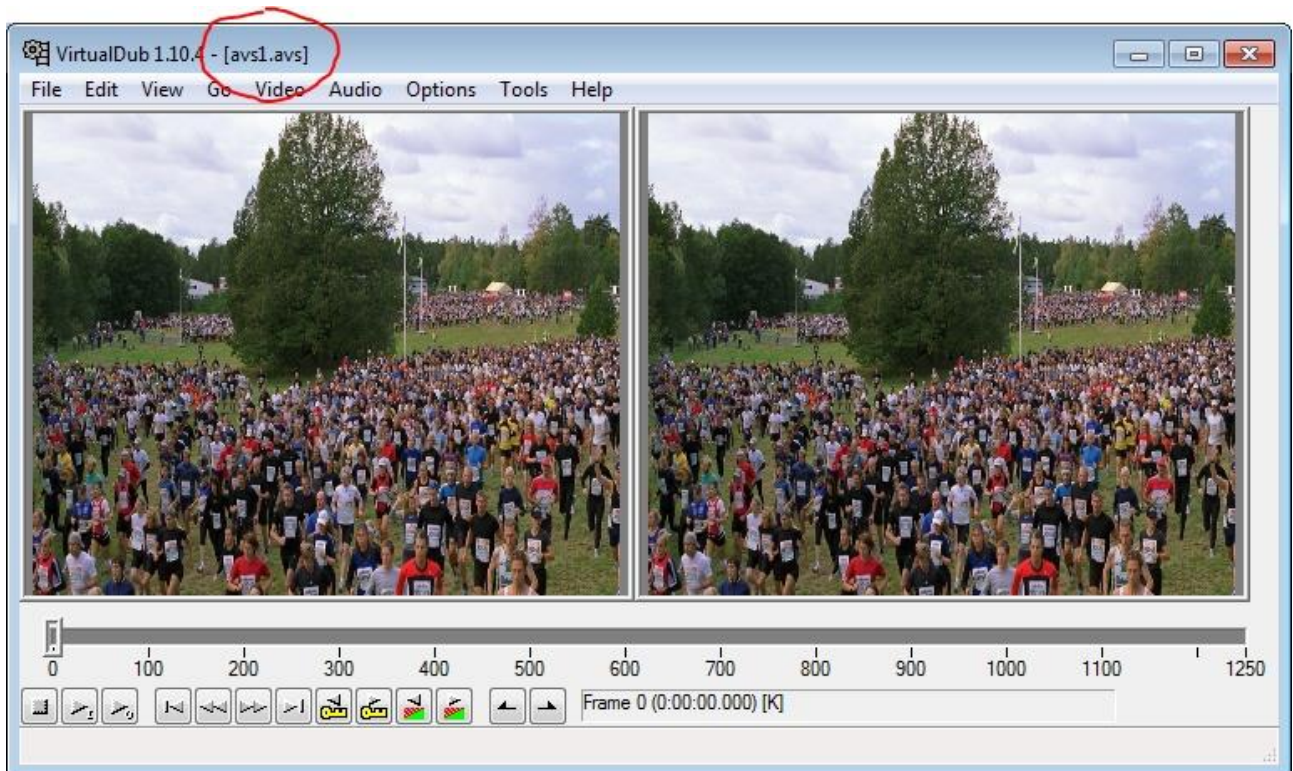
- > Créez un nouveau document texte que vous nommerez : « AVS1.avs »
- > Ouvrez-le avec notepad et tapez le chemin vers votre fichier.

```
AviSource("E:\EBU_5seq_NonCompress_576i.avi")
```

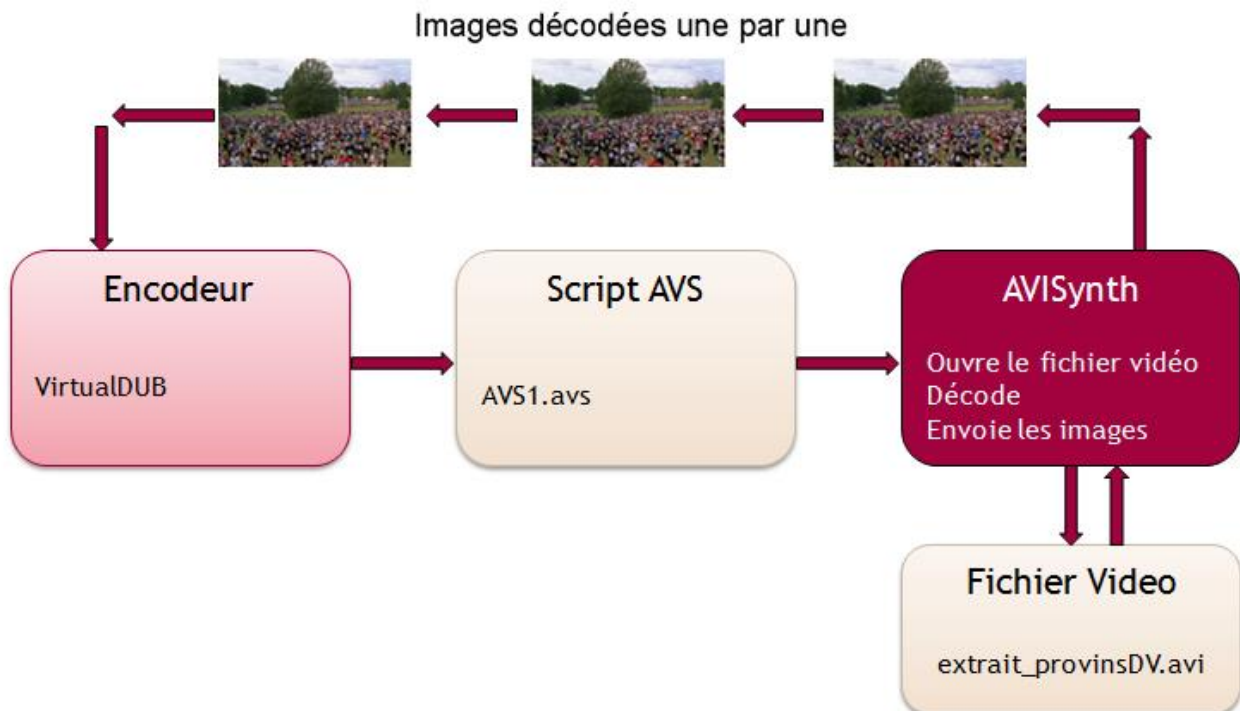
- > Ouvrons par exemple Virtualdub



- > ->File -> « Open Video File »
- > Choisissez le script AVS1.avs



C'est bien le fichier AVS1.avs qui s'est ouvert. Pourtant nous voyons les images du fichier extrait_provinsDV.avi



■ Remarque

Ce premier script peut présenter le désavantage de rajouter une couche à notre utilisation. Cependant, si nous prenons notre exemple VirtualDUB ne sait ouvrir que des

fichiers AVI. Si nous avons un fichier non compris par VirtualDUB mais compris par AVIsynth, nous pourrions remplacer la commande "AVIsynth" par "DirectShowSource" et ainsi utiliser d'autres types de fichiers dans VirtualDUB.
Nous allons voir qu'AVIsynth possède d'autres fonctionnalités très utiles.

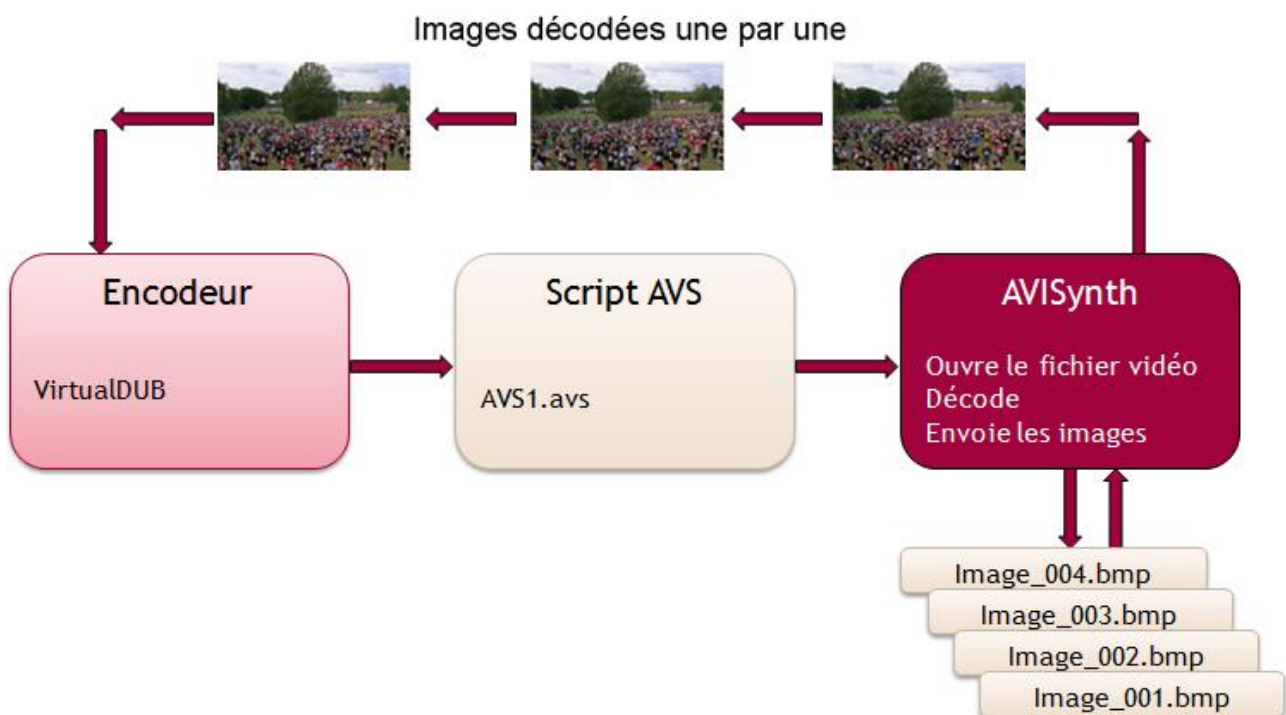
5.2.3. Convertir une série d'images en vidéo

Ma source vidéo n'est pas obligatoirement une vidéo !!! Elle peut être une suite d'images
Créez un fichier AVS (par exemple : avs2.avs) et ouvrez-le avec NotePad.

```
ImageSource("E:\Medias_TD_VideoenLigneDeCommande\Images\course-%04d.bmp",
start=0001, end=0250, fps=25)
ConvertToYUY2(interlaced=true, matrix="rec709")
```

- > ImageSource signifie que la source est une suite d'images
- > %04d signifie que la numérotation du nom du fichier source est sur 4 chiffres
- > Start=0001 : la première image
- > End=0250 : la dernière image
- > Fps=25 : nombre d'images par seconde
- > ConvertToYUY2 Convertit en 4:2:2 (entrelacé, colorimétrie « rec709 »)
- > (interlaced=true) : en entrelacé
- > matrix="rec709" : colorimétrie « rec709 » c'est à dire HD

Ouvrir ce fichier Avs dans VirtualDub



VirtualDub recevra une vidéo créée à partir de plusieurs images fixes. Grâce à ce script AVS nous pouvons ainsi transformer une suite d'images et l'utiliser comme une vidéo dans de nombreux logiciels.

Nous commençons également à toucher la puissance de cet outil qui nous permet de donner une fréquence image (25), de transformer l'espace colorimétrique pour passer de RVB en YUV adapté à la haute définition, et d'entrelacer les images.

5.2.4. Un Editeur dédié : AVSP

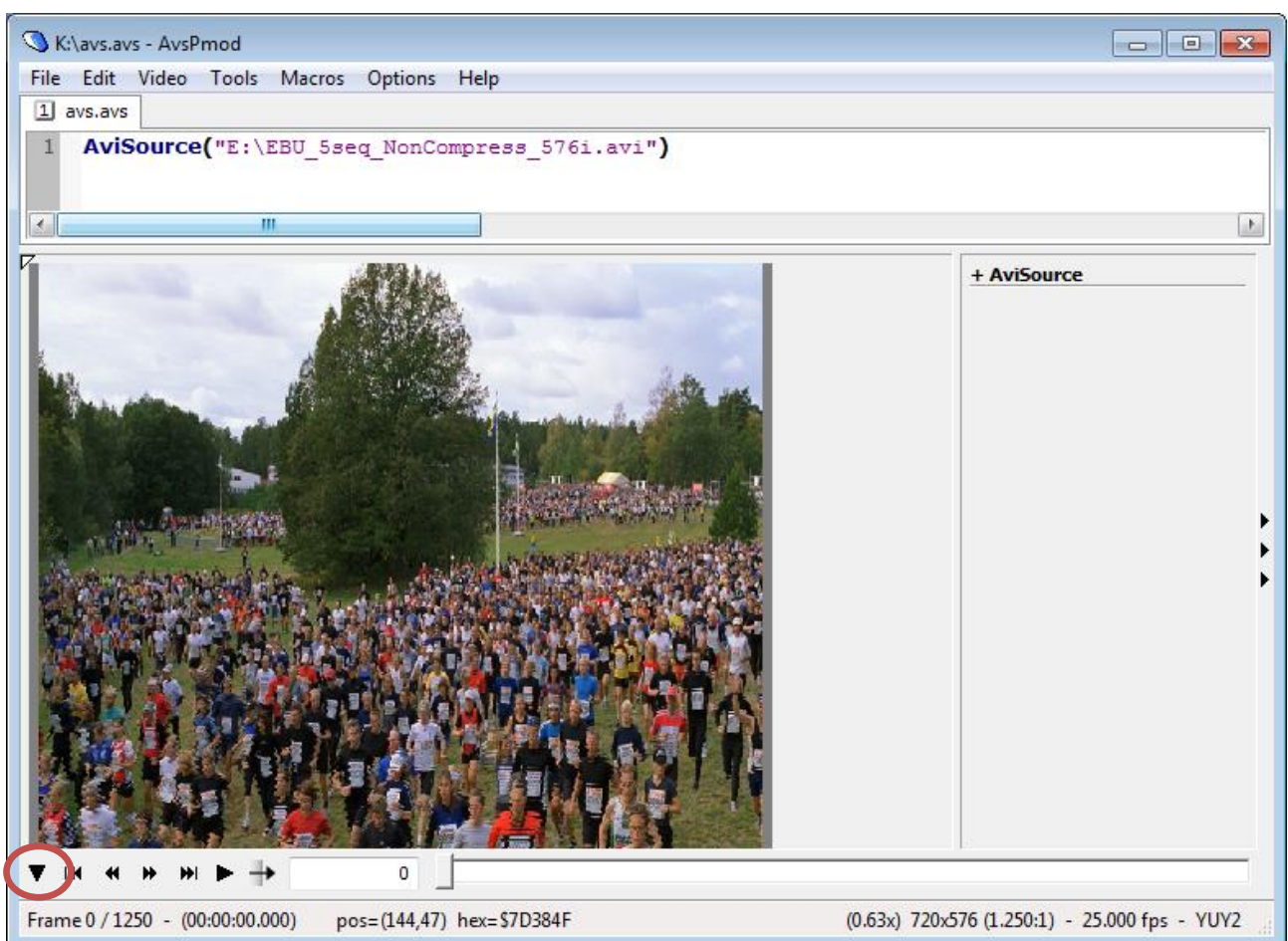
Notepad n'est pas un éditeur de texte très pratique, et pas adapté aux scripts AVS. Voici AVSP. C'est un éditeur dédié AVIsynth. Il connaît certaines commandes AVIsynth, mais surtout, il permet de faire une prévisualisation des scripts.

Téléchargeable à cette adresse :

<http://avisynth.org/qwerpoi/index.html>

La encore, des variantes existent. Vous pouvez trouver AVSPmod qui permet de travailler en 64b :

<http://avspmod.github.io/>



On peut avoir une prévisualisation, ce qui permet de vérifier que le script est correct, de visualiser le rendu des filtres grâce à la petite flèche en bas à gauche de la fenêtre.

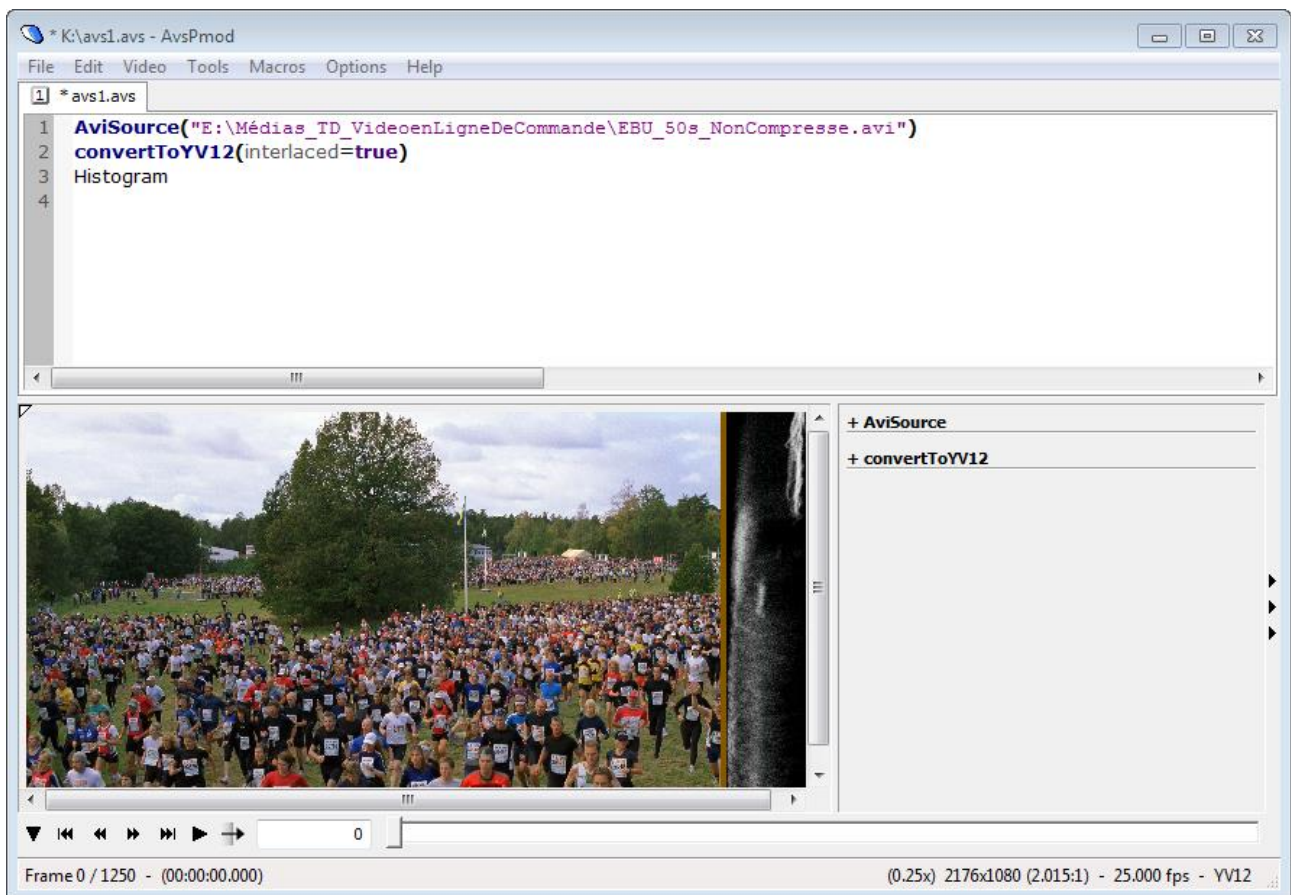
5.2.5. Usage des filtres : Histogram

Toute la puissance d'AVIsynth réside dans l'usage des filtres
Prenons un exemple simple :

```
AviSource("E:\Medias_TD_VideoenLigneDeCommande\EBU_50s_NonComprimee.avi")
convertToYV12(interlaced=true)
Histogram
```

- > AviSource ouvre le fichier
- > convertToYV12 fait un sous échantillonnage en 4:2:0 (beaucoup de filtres d'AVIsynth ne fonctionnent qu'en 4:2:0)
- > Histogram ajoute un « histogramme » à droite du clip.

Le résultat est une nouvelle image avec un histogramme incrusté. Cette nouvelle image, générée par AVIsynth, peut maintenant être compressée par un encodeur (exemple : « virtualDub »).



5.2.6. Exercice : Incrustation d'un logo



Enseignement
Formation
Recherche

L'objectif est de placer sur une vidéo un logo donné (ce logo pourrait être une petite vidéo) :



Enseignement
Formation
Recherche

Logo de l'INA

Nous pourrions gérer la transparence grâce à une image alpha (image en niveaux de gris : noir transparent, blanc opaque)



Nous allons utiliser le filtre **overlay** :

```
overlay(video_originale, logo, X, Y, mask, 1)
```

Le filtre overlay définit les paramètres pour la superposition du logo, dans l'ordre on trouve :

- > le clip de base (vidéo_originale)
- > l'image superposée (logo, peut être une vidéo)
- > le placement en X à partir du coin en haut à gauche
- > le placement en Y
- > l'image utilisée comme masque
- > l'opacité (de 0 à 1, 100% opaque dans ce cas)



5.2.7. Exemple : filtres géométriques

Crop	Rogne l'image.
CropBottom	Rogne les pixels en excès au bas de l'image.
Letterbox	Letterbox est une combinaison entre AddBorder et Crop. Au lieu de rogner l'image comme Crop, Letterbox remplace les pixels sélectionnés par du noir.
ReduceBy2	Réduit la taille de l'image par 2.
HorizontalReduceBy2 / VerticalReduceBy2	Réduit la longueur (horizontal) ou la hauteur (vertical) de l'image par 2.
BicubicResize / BilinearResize / GaussResize / LanczosResize / Lanczos4Resize / PointResize / Spline16Resize / Spline36Resize	Ces filtres de redimensionnement modifient la taille de l'image à une valeur choisie. Ces différents filtres utilisent chacun un algorithme de redimensionnement différent.
FlipHorizontal / FlipVertical	Inverse l'image horizontalement ou verticalement (la gauche devient la droite, le haut devient le bas)
TurnLeft / TurnRight / Turn180	Tourne l'image de 90 degrés dans le sens inverse des aiguilles d'une montre / 90 degrés dans le sens des aiguilles d'une montre.
StackHorizontal / StackVertical	Prend plusieurs clip en entrée et restitue un clip où ils sont les uns à côté des autres horizontalement ou verticalement.

5.2.8. Exemple : filtres d'entrelacement

AssumeFrameBased / AssumeFieldBased	Force une vidéo à être basée sur les frames ou sur les champs.
AssumeBFF / AssumeTFF	Force l'ordre des champs (champs du bas/haut en premier).
Bob	Désentrelacement Bob basique (double le framerate).
ComplementParity	Inverse le champ du haut et le champ du bas.
DoubleWeave	Ce filtre agit comme Weave , à part qu'il produit un clip avec le double de frames en combinant les paires de champs des frames paires et impaires.
Pulldown	Ce filtre permet de télécinéer votre vidéo.
SeparateFields	Force une vidéo basée sur les frames à se baser sur les champs en séparant chaque image en deux champs (le champ du haut et le champ du bas).
SwapFields	Swaps the two fields in an interlaced frame.

5.2.9. Exercice Downconversion

Convertir la séquence HD 1080i EBU_50s_NonComprimee.avi en SD.

Quelles sont les étapes ?

- 1) ouvrir le fichier
- 2)
- 3)
- 4)

et le script :

5.2.10. SplitScreen

Réaliser un écran partagé avec 2 fichiers HD :
EBU_50s_NonComprimee.avi et EBU_50s_X264.mp4
Rajouter pour chaque partie de l'écran un texte.

```
Clip1=aviSource("E:\Medias_TD_VideoenLigneDeCommande\EBU_50s_NonComprimee.avi")
Clip2=directshowSource ("k:\EBU_50s_X264.mp4")
videogauche=Clip1.crop(0,0,960,1080).ConvertToYUY2
videodroite=Clip2.crop(960,0,0,1080).ConvertToYUY2
Composition=stackhorizontal(videogauche, videodroite)
Composition=subtitle (composition, "Clip de gauche", x=480,
y=50,font="verdana", size=48, text_color=$ffffff, halo_color=$000000,
align=2)
subtitle (composition, "Clip de droite", x=1440, y=50,font="verdana",
size=48, text_color=$ffffff, halo_color=$000000, align=2)
```

5.3. Pour aller plus loin

Si les filtres d'AVIsynth ne suffisent pas, il est possible d'importer des filtres externes avec la commande

LoadPlugin ("NomDuPlugin" [, ...])

Les Filtres externes validés par AVIsynth se trouvent à cette page

http://avisynth.nl/index.php/External_filters

Si les filtres d'AVIsynth ne suffisent toujours pas, il est possible d'importer des filtres VIRTUALDub. Certains peuvent être très puissants.

Voici un site recensant de nombreux filtres pour VirtualDUB :

<http://www.infognition.com/VirtualDubFilters/>

Mettez « MSU_Cartoonizer.vdf » au même endroit que votre script AVS.

```
LoadVirtualDubPlugin("K:\MSU_Cartoonizer.vdf", "MSU_Cartoonizer", 0)
clip=AviSource("E:\Medias_TD_VideoenLigneDeCommande\EBU_50s_NonCompresse.
avi", false, "RGB24")
clip.ConvertToRGB32.MSU_Cartoonizer(0, 1, 5, 5, 2, 2, 1, 5, 5, 5, 5, 1)
```

Regardez le résultat !

La dernière ligne correspond aux paramètres de MSU_Cartoonizer :

```
clip.ConvertToRGB32.MSU_Cartoonizer(0, ShowEdges, EdgeAmount,
EdgeBrightness, EdgeThickness, EdgeAccuracy, UseFaceDetection, BrushSize,
BrushWetness, ColorRandom, ColorSaturation, ProcessEachNframe)
```

- ShowEdges - sets visibility of the edges. ShowEdges can be 0 - invisible edges and 1 - visible edges.
- EdgeAmount - integer between 0 and 10. Sets the amount of edges.
- EdgeBrightness - integer between 0 and 10. Sets the edge brightness. Default value is 5.
- EdgeThickness - integer between 1 and 4. Sets the edge thickness, 1 - thin edges and 4 - thick edges.
- EdgeAccuracy - integer between 1 and 4. Sets the stroke accuracy, 1 - maximum edge accuracy and 4 - minimum edge accuracy.
- UseFaceDetection - allows improved edge rendering for Faces. UseFaceDetection can be 0 - do not use improved edge rendering for Faces, 1 - use improved edge rendering for Faces.
- BrushSize - integer between 1 and 10. Sets the size of a brush.
- BrushWetness - integer between 0 and 10. Sets the brush wetness.
- ColorRandom - integer between 0 and 10. Sets the color randomization.
- ColorSaturation - integer between 0 and 10. Sets the color saturation. Default value is 5.
- ProcessEachNframe - integer between 1 and 10. Sets the amount of processed frames.

6. Automatisation

L'avantage de la ligne de commande est de pouvoir automatiser le traitement de nos vidéos. Nous allons voir les bases de l'écriture de script sur PC qui permettront d'automatiser nos traitements.

6.1. Les fichiers BAT

L'avantage de la ligne de commande est la possibilité d'automatiser ce que l'on vient de voir. Nous allons utiliser des fichiers BAT

Le fichier BAT est un fichier texte que l'on peut éditer avec notepad

Le fichier bat est une suite de commandes DOS que l'on enchaîne.

FFmpeg, une fois installé ou situé dans le même dossier que le .bat, devient une commande DOS

6.2. Automatiser le traitement d'un dossier

Nous voulons traiter tous les fichiers se trouvant dans un dossier de la même façon. Par exemple, convertir tous les fichiers AVI en MP4

Créer 2 dossiers, un nommé "AEncoder" (par exemple) et le second "Resultat"

Nous allons utiliser une variable %%a

```
cd Aencoder
for %%a in (*.avi) do ffmpeg.exe -i "%%a" -c:v mpeg4 -vb 1200K -c:a
libmp3lame -b:a 128K -y "..\Resultat\%%~na_ENCODE.avi"
```

Sauvegarder ce texte en script.bat par exemple

- Remarque

%%a renvoi le nom du fichier en cours

%%~na renvoie le nom du fichier sans son extension, ce qui permet de renommer simplement le fichier de sortie.

6.3. Automatiser la création de scripts AVS

On peut aller encore plus loin et automatiser aussi l'utilisation de AVIsynth. Pour cela il faut créer des fichiers avs à partir de notre script grâce à la commande echo qui permet d'écrire dans un fichier.

Regardons l'exemple suivant :

Convertir un fichier MTS (caméscope AVCHD) en DV (vidéo standard en 16/9)

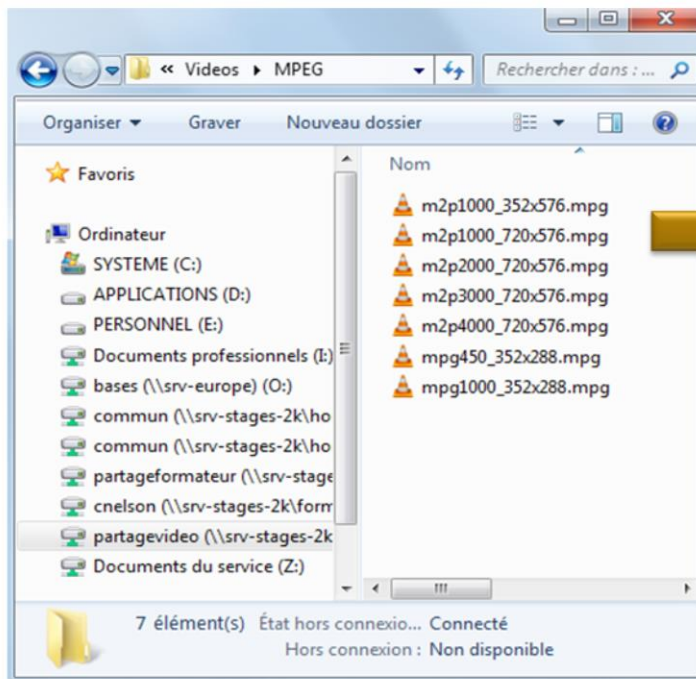
```
cd AEncoder
rem preparation automatisée du fichier AVS
```

```
for %%a in (*.mts) do @echo DirectShowSource("%%a", audio=true)
>>"%%~na.avs"
for %%a in (*.mts) do @echo Lanczos4Resize(720,576) >> "%%~na.avs"
rem encodage FFMPEG
for %%b in (*.avs) do ffmpeg.exe -i "%%b" -aspect 16:9 -vcodec dvvideo
-acodec copy -y "..\Resultat\%%~nb_DV.avi"
del *.avs
```

La commande echo texte >> script.avs permet de mettre "texte" dans un fichier nommé script.avs

6.4. Automatiser : Drag and drop

Lorsque l'on fait un glisser déposer d'un fichier vidéo "mavideo.avi" sur un fichier batch : "information.bat" par exemple, c'est comme si on lançait en ligne de commande le batch avec un paramètre :



Glisser-déposer



Information.bat

```
information.bat mavideo.avi
```

Il faut maintenant pouvoir récupérer ce paramètre comme étant le fichier à traiter. La variable qui permet de récupérer ce paramètre est %1 (%2, %3 s'il y avait plusieurs paramètres)

Créons maintenant un fichier texte que nous nommerons "information.bat"

Avec un clic droit -> modifier, on va pouvoir l'éditer :

```
ffmpeg.exe -i %1
pause
```

Sauvegardez puis glissez un fichier vidéo sur l'icône de "information.bat"
La commande pause est indispensable sinon la fenêtre se ferme.

```
C:\Windows\system32\cmd.exe
E:\temp>ffmpeg.exe -i E:\temp\400k_withouthblur.avi
FFmpeg version SVN-r26400, Copyright (c) 2000-2011 the FFmpeg developers
built on Jan 18 2011 04:07:05 with gcc 4.4.2
configuration: --enable-gpl --enable-version3 --enable-libgsm --enable-libvorbis
--enable-libtheora --enable-libspeex --enable-libmp3lame --enable-libopenjpeg
--enable-libschroedinger --enable-libopencore_amrwb --enable-libopencore_amrnb
--enable-libvpx --disable-decoder=libvpx --arch=x86 --enable-runtime-cpudetect --
enable-libxvid --enable-libx264 --enable-librtmp --extra-libs='-lrtmp -lpolarss
l -lws2_32 -lwinmm' --target-os=mingw32 --enable-avisynth --enable-w32threads --
cross-prefix=i686-mingw32- --cc='ccache i686-mingw32-gcc' --enable-memalign-hack

libavutil      50.36. 0 / 50.36. 0
libavcore       0.16. 1 /  0.16. 1
libavcodec     52.108. 0 / 52.108. 0
libavformat     52.93. 0 / 52.93. 0
libavdevice     52. 2. 3 / 52. 2. 3
libavfilter      1.74. 0 /  1.74. 0
libswscale      0.12. 0 /  0.12. 0
[mpeg4 @ 028b1fa0] Invalid and inefficient vfw-avi packed B frames detected
Seems stream 0 codec frame rate differs from container frame rate: 30000.00 (300
00/1) -> 25.00 (25/1)
Input #0, avi, from 'E:\temp\400k_withouthblur.avi':
  Duration: 00:00:22.08, start: 0.000000, bitrate: 506 kb/s
    Stream #0.0: Video: mpeg4, yuv420p, 360x270 [PAR 1:1 DAR 4:3], 25 fps, 25 tbn
r, 25 tbn, 30k tbc
    Stream #0.1: Audio: mp3, 48000 Hz, 2 channels, s16, 96 kb/s
At least one output file must be specified

E:\temp>pause
Appuyez sur une touche pour continuer...
```

Ces informations, on peut les exporter vers un fichier. Il faut remplacer le script par :

```
ffmpeg.exe -i %1 >> InfoFichier.txt
```

6.5. Convertir le batch en exe

On peut trouver sur le net de nombreux convertisseurs bat en exécutable, par exemple :

<http://www.f2ko.de/programs.php?lang=en&pid=b2e>

Avoir un exécutable permet de rendre la fenêtre dos invisible et présente d'autres avantages pour "finaliser" un produit.

6.6. Ajout d'une icône

L'icône par défaut d'un .bat est



Il peut être plus agréable pour vos utilisateurs d'identifier simplement la fonctionnalité du script. Par exemple



Pour cela, concevoir une image en 256x256 dans un logiciel d'édition d'image. Il est possible de travailler sur une image plus grande mais elle sera convertie en 256x256. Exporter en PNG

Aller sur <http://www.convertico.com/>, et sélectionner ce fichier PNG

Cliquer su GO

SELECT FILE FROM URL

http://

SELECT FILE FROM YOUR COMPUTER

BROWSE

Télécharger le fichier ainsi converti



Créer un raccourci du script (maintenir la touche Alt)

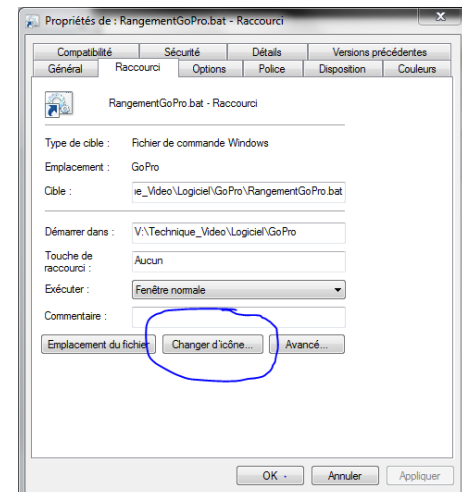
	RangementGoPro.bat	12/12/2014 16:02	Fichier de cor
	RangementGoPro.bat - Raccourci	12/12/2014 16:01	Raccourci

Faire un clic droit sur ce raccourci et choisir propriétés

Cliquer sur « Changer d'icône » en bas

Cliquer sur « Parcourir... »

Choisir le fichier icône que nous venons de réaliser



7. Annexes

7.1. Corrections d'exercices

> Exercice 4.2.5

```
ffmpeg -i input.avs -c:v libx264 -preset slow -b:v 7.5M video_finale.mp4
```

> Exercice 4.2.9

```
ffmpeg.exe -i k:\10s_HD_Non_Compressee.avi -c:v libx265 -preset slow -x265-params qp=22 video_H265.mp4
```

> Exercice 4.3.3

```
ffmpeg -i k:\10s_HD_Non_Compressee.avi -i k:\ina2.png -filter_complex overlay=W-w output.mp4
```

> Exercice 5.2.6

Voici une façon d'incruster le logo :

```
video_originale=AviSource("E:\Medias_TD_VideoenLigneDeCommande\EBU_50s_NonComprese.avi")
logo = ImageSource("E:\Medias_TD_VideoenLigneDeCommande\ina2.png")
mask = ImageSource("E:\Medias_TD_VideoenLigneDeCommande\ina2_mask.png")
video_resultat = overlay(video_originale, logo, 1600, 100, mask, 1)
video_resultat
```

> Exercice 5.2.9

Une façon de faire peut être celle-ci.

```
aviSource
("E:\Medias_TD_VideoenLigneDeCommande\EBU_50s_NonComprese.avi")
SeparateFields()
lanczosresize(720, 288)
Weave
```

Le principe est de séparer les 2 trames, de redimensionner ces trames, puis les réassembler ensemble, afin de garder l'entrelacement en SD.

7.2. Les codes informatiques pour les formats de chrominances

Nom	Quantification	Chrominance
RVB24	8b	4 : 4 : 4
RVB32	8b	4 : 4 : 4 canal Alpha
YV24	8b	4 : 4 : 4
YUY2 UYUY YV16	8b	4 : 2 : 2
YV12 IYUV	8b	4 : 2 : 0
YV411	8b	4 : 1 : 1

Y8	8b	4 :0 :0
----	----	---------

Voici les différentes dénominations permettant de faire le lien entre les mondes informatique et vidéo

7.3. Quelques commandes DOS

Astuce : `cd c:\prog*` évite de taper tout le nom du répertoire

APPEND Permet à des applications d'ouvrir des fichiers se trouvant dans des répertoires comme s'ils étaient dans le répertoire en cours.

AT Planifie l'exécution de commandes ou programmes sur un ordinateur.

ATTRIB Affiche ou modifie les attributs d'un fichier.

BREAK Active ou désactive le contrôle étendu de CTRL+C.

CACLS Affiche ou modifie les listes de contrôles d'accès aux fichiers.

CALL Appelle un fichier de commandes depuis un autre fichier de commandes.

CLS Efface l'écran.

CMD Lance une nouvelle instance de l'interpréteur de commandes de Windows 2000.

COLOR Modifie les couleurs du premier plan et de l'arrière plan de la console.

COMP Compare les contenus de deux fichiers ou groupes de fichiers.

COPY Copie un ou plusieurs fichiers.

DATE Affiche ou modifie la date.

DEL Supprime un ou plusieurs fichiers.

DIR Affiche la liste des fichiers et des sous-répertoires d'un répertoire.

ECHO Affiche des messages à l'écran ou active/désactive l'affichage des commandes.

ENDLOCAL Stoppe la localisation des modifications de l'environnement dans un fichier de commandes.

ERASE Supprime un ou plusieurs fichiers.

EXIT Quitte l'interpréteur de commandes (CMD.EXE).

EXPAND Décompresse un ou plusieurs fichiers compressés.

FC Compare deux fichiers ou groupes de fichiers, et affiche les différences entre eux.

FIND Cherche une chaîne de caractères dans un ou plusieurs fichiers.

FINDSTR Cherche des chaînes de caractères dans un ou plusieurs fichiers.

FOR Exécute une commande sur chaque fichier d'un groupe de fichiers.

FTP Transfère des fichiers vers et à partir d'un ordinateur exécutant un service FTP

GOTO Poursuit l'exécution d'un fichier de commandes à une ligne identifiée par une étiquette.

HELP Affiche des informations sur les commandes de Windows 2000.

IF Effectue un traitement conditionnel dans un fichier de commandes.

IPCONFIG Affiche la configuration IP de Windows 2000. `Cmd /k Ipconfig` peut-être effectué à partir d'Exécuter.

LABEL Crée, modifie ou supprime le nom de volume d'un disque.

MD Crée un répertoire.

MEM Affiche les quantités de mémoire utilisée et libre sur votre système.

MKDIR Crée un répertoire.

MORE Affiche la sortie écran par écran.

MOVE Déplace un ou plusieurs fichiers d'un répertoire à un autre

PATH Affiche ou définit le chemin de recherche des fichiers exécutables.

PAUSE Interrompt l'exécution d'un fichier de commandes et affiche un message.

PING Affiche les informations de connexion à une autre machine.

PRINT Imprime un fichier texte.

PROMPT Modifie l'invite de commande de Windows 2000.

RD Supprime un répertoire.

RECOVER Récupère l'information lisible d'un disque défectueux.

REM Insère un commentaire dans un fichier de commandes ou CONFIG.SYS.

REN Renomme un ou plusieurs fichiers.

RENAME Renomme un ou plusieurs fichiers.

REPLACE Remplace des fichiers.

RMDIR Supprime un répertoire.
 SORT Trie les éléments en entrée.
 START Lance une fenêtre pour l'exécution du programme ou de la commande.
 TIME Affiche ou définit l'heure de l'horloge interne du système.
 TITLE Définit le titre de la fenêtre pour une session CMD.EXE.
 TYPE Affiche le contenu d'un fichier texte.
 VOL Affiche le nom et le numéro de série du volume.
 XCOPY Copie des fichiers et des arborescences de répertoires.

7.4. Les Noms de variable « fichiers » en batch

La table ci dessous permet de selectionner des informations "passées"

Parameter	Description
%1	Le nom de fichier
%~f1	Donne le chemin absolu complet
%~d1	Donne la lettre du driver sur lequel est %1.
%~p1	Donne le chemin de %1.
%~n1	Donne le nom sans l'extension
%~x1	Donne l'extension
%~s1	Donne le nom court du fichier

Des associations de paramètres :

Parameter	Description
%~dp1	Associe la lettre du driver et le chemin
%~sp1	Le chemin court
%~nx1	Le nom et l'extension

Pour voir tous ces paramètres en action :

```
@echo off<br>
echo fully qualified name %~f1<br>
echo drive %~d1<br>
echo path %~p1<br>
echo filename %~n1<br>
echo file extension %~x1<br>
echo short filename %~sn1<br>
echo short file extension %~sx1<br>
echo drive and directory %~dp1<br>
echo filename and extension %~nx1
```

Par exemple, avec un fichier se trouvant dans c:\temp\longfilename.long voici le résultat :

```
fully qualified name c:\TEMP\longfilename.long<br>
drive c:<br>
path \TEMP\<br>
filename longfilename<br>
file extension .long<br>
short filename LONGFI~1<br>
short file extension .LON<br>
```

```
drive and directory c:\TEMP\  
filename and extension longfilename.long
```

8. Table des matières

1. Objectifs :	2
2. Outils nécessaires :	2
2.1. logiciels	2
2.2. Système d'exploitation	2
3. Introduction aux lignes de commandes	2
3.1. Pourquoi utiliser des lignes de commande	3
4. Encodeur en lignes de commande	4
4.1. Présentation de FFMPEG	4
4.1.1. Compilation	4
4.1.2. Installation	5
4.1.3. L'interface en ligne de commande	5
4.1.4. Structure de FFMPEG	6
4.1.5. La documentation de FFmpeg	6
4.2. Syntaxe de FFMPEG (Coder et Muxer)	7
4.2.1. Exemple : Obtenir des informations sur la vidéo	7
4.2.2. Exemple : Convertir en WebM	8
4.2.3. Réglage du débit	9
4.2.4. Choix du Codec (Convertir en format DV)	9
4.2.5. Choix du Codec (Encoder en h264)	10
4.2.6. Choix du Codec (Video non compressé)	11
4.2.7. Forcer l'écriture	11
4.2.8. Déwrapper-Wrapper	11
4.2.9. Exercice Encoder en H265	11
4.2.10. Extraire des images d'une vidéo	12
4.3. Syntaxe de FFMPEG (Filter)	12
4.3.1. Quelques options vidéo :	12
4.3.2. Exemple : resize	13
4.3.1. Video filter	13
4.3.2. Encoder un nombre d'images en vidéo	13
4.3.3. Insérer un logo	13
4.4. Pour aller plus loin	14
4.4.1. Usage de FFplay	14
4.4.2. FFprobe	15
4.4.3. FFserver	15
4.4.4. FFMBC	15
5. Le serveur d'images	16

5.1.	Qu'est-ce qu'un FrameServer ?	16
5.1.1.	Fonctionnement d'un FrameServer :.....	16
5.1.2.	Exemple de FrameServer : AVISynth	16
5.1.3.	Installation d'AVISynth	16
5.2.	Utilisation d'AVISynth	17
5.2.1.	Script AVISynth.....	18
5.2.2.	1 ^{er} script : ouvrir une source vidéo	19
5.2.3.	Convertir une série d'images en vidéo	21
5.2.4.	Un Editeur dédié : AVSP	22
5.2.5.	Usage des filtres : Histogram	23
5.2.6.	Exercice : Incrustation d'un logo	23
5.2.7.	Exemple : filtres géométriques.....	25
5.2.8.	Exemple : filtres d'entrelacement	25
5.2.9.	Exercice Downconversion.....	26
5.2.10.	SplitScreen.....	26
5.3.	Pour aller plus loin	26
6.	Automatisation	28
6.1.	Les fichiers BAT	28
6.2.	Automatiser le traitement d'un dossier.....	28
6.3.	Automatiser la création de scripts AVS	28
6.4.	Automatiser : Drag and drop	29
6.5.	Convertir le batch en exe.....	30
6.6.	Ajout d'une icône	30
7.	Annexes	32
7.1.	Corrections d'exercices	32
7.2.	Les codes informatiques pour les formats de chrominances	32
7.3.	Quelques commandes DOS	33
7.4.	Les Noms de variable « fichiers » en batch	34
8.	Table des matières	36