

## Sous-titrage stéréoscopique

<b>Introduction .....</b>	<b>1</b>
<b>Le sous-titrage .....</b>	<b>2</b>
<b>Utilisation du filtre <i>subtitler</i> pour <i>VirtualDub</i>.....</b>	<b>2</b>
<b>Écriture du fichier de sous-titre .....</b>	<b>5</b>
<b>Utilisation de <i>Stereoscopic Player</i> .....</b>	<b>8</b>
<b>Liens internet .....</b>	<b>11</b>

### Introduction

Si, d'ordinaire, nous réalisons nos diaporamas en français, pour toucher un public plus vaste, il peut être utile de produire des versions dans d'autres langues. Par exemple pour une diffusion sur Internet ou pour rendre plus accessible un programme destiné à être présenté lors d'un congrès ISU.

Dans un diaporama classique, un certain nombre d'éléments auront besoin d'être traduits. On peut les grouper en deux catégories :

- Titres, génériques, légendes,... donc des éléments écrits.
- Commentaires parlés, donc des éléments sonores.

L'auteur du diaporama est généralement à l'origine des éléments textuels et il lui sera donc (relativement) facile de générer ces visuels dans une autre langue ou bien de faire une version bilingue. Nous nous intéresserons donc ici à la problématique découlant des commentaires parlés.



*Exemple de titrage bilingue*

Pour les commentaires parlés on peut envisager deux approches :

- avoir les commentaires prononcés dans la langue cible, ils remplaceront alors, dans la bande son, les commentaires lus dans la langue d'origine.
- ou bien afficher la traduction des commentaires comme des sous-titres.

Dans les deux cas, la première opération est bien sûr la traduction en elle-même dans la langue cible. Même avec de bonnes connaissances de cette langue, il sera probablement souhaitable de faire vérifier le texte par une personne l'ayant comme langue maternelle. De même le premier cas nécessite une personne ayant une bonne prononciation et diction dans la langue considérée. Nous nous restreindrons donc dans cet exposé à la réalisation de sous-titres.

## Le sous-titrage

Le principe du sous-titrage est de présenter un texte en surimpression sur l'image à un moment précis et pour une durée déterminée. Pour rendre le texte plus lisible on peut lui donner un fond opaque mais ça n'est généralement pas très esthétique. Il est préférable d'utiliser un fond transparent et d'utiliser une bordure des caractères contrastante de manière à ce que le texte reste lisible, qu'il soit sur un fond clair ou un fond sombre.

Les techniques de sous-titrage peuvent se classer en deux catégories : incrustation (*hardsubs* en anglais) où le texte est intégré dans les images et ne pourra donc pas être enlevé et la superposition (*softsubs*) où le texte est ajouté (ou non) par le logiciel de lecture vidéo au moment même de la lecture.



*Exemple de sous-titre à fond transparent.  
Les caractères sont jaunes avec une bordure noire.  
Notez également l'ombrage translucide.*

## Utilisation du filtre *subtitler* pour *VirtualDub*

Nous allons utiliser l'outil gratuit *VirtualDub* qui est une sorte de couteau suisse du traitement vidéo pour réaliser un sous-titrage par incrustation.

Les fonctionnalités de *VirtualDub* peuvent être étendues par l'usage de *filtres*. Un filtre ajoute un traitement spécifique au logiciel : redimensionnement, traitement de couleur,... Le filtre qui nous intéresse ici s'appelle *subtitler* (soit « sous-titreur »). On le trouvera sur le même site que *VirtualDub*.

La configuration des filtres simples s'effectue par une fenêtre de paramétrage. Le filtre *subtitler* a besoin de plus d'informations : les textes eux-mêmes, mais aussi leur position, instant d'affichage, durée,... Pour cette raison le filtre utilise un fichier de sous-titre externe au format SSA (*Sub Station Alpha*). Il existe des éditeurs de fichiers SSA facilitant leur rédaction mais on peut également se contenter du bloc-notes de Windows dans la plupart des cas.

*Subtitler* est fourni avec un fichier d'exemple assez complet démontrant un grand nombre des possibilités du filtre (positionnement, gestion des collisions, mode karaoké,...). Il peut servir de base pour écrire son propre fichier de sous-titre.

### **En 2D oui, mais en 3D ?**

Pour la suite de l'exposé nous allons utiliser un diaporama sous la forme d'une vidéo AVI en mode côte-à-côte (donc en double largeur). Chargeons cette vidéo dans *VirtualDub*, ajoutons (menu *Video > Filters*) le filtre *subtitler* et, dans ce filtre chargeons le fichier d'exemple.

Ça fonctionne, oui mais en 2D seulement ! Avec une image stéréoscopique, le sous-titre s'inscrit sur l'ensemble du couple ce qui ne va pas du tout. Ce qu'il nous faut c'est que le texte soit dupliqué sur les vues gauche et droite. Comment faire alors ? En réfléchissant un peu on entrevoit deux solutions possibles :

- Séparer la vidéo stéréoscopique en deux vidéos gauche et droite, sous-titrer chacune de ces vidéos puis réassembler les deux en une vidéo stéréoscopique côte-à-côte.

- Jouer sur la taille des marges dans le fichier de sous-titre. On ajoute un deuxième filtre *subtitler*. Sur le premier *subtitler*, on règle la marge droite suffisamment large pour que le texte ne déborde pas sur l'image droite et sur le deuxième, on jouera sur la marge gauche pour repousser le texte dans l'image droite du couple.



*Le sous-titrage 2D ne fonctionne pas bien sur une image stéréoscopique !*

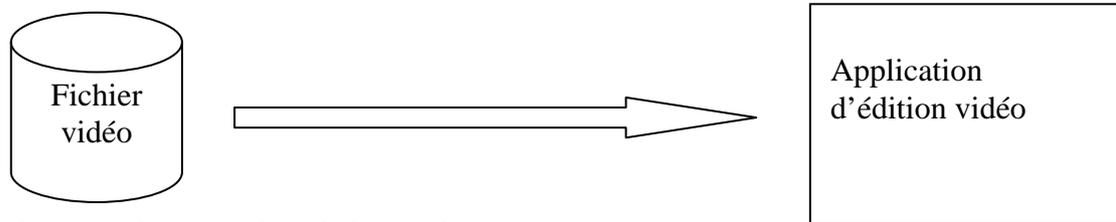
Ces solutions ont quand même des inconvénients. La première multiplie les opérations et les fichiers vidéo intermédiaires qui, si on souhaite préserver la qualité d'image, vont être de très grande taille. La seconde est peu pratique, les marges n'étant pas spécifiées en pourcentage de la taille de l'image.

Je propose donc une autre approche utilisant un autre outil de traitement vidéo très pratique (et gratuit !) : *AviSynth*.

### **Utilisation du logiciel *AviSynth***

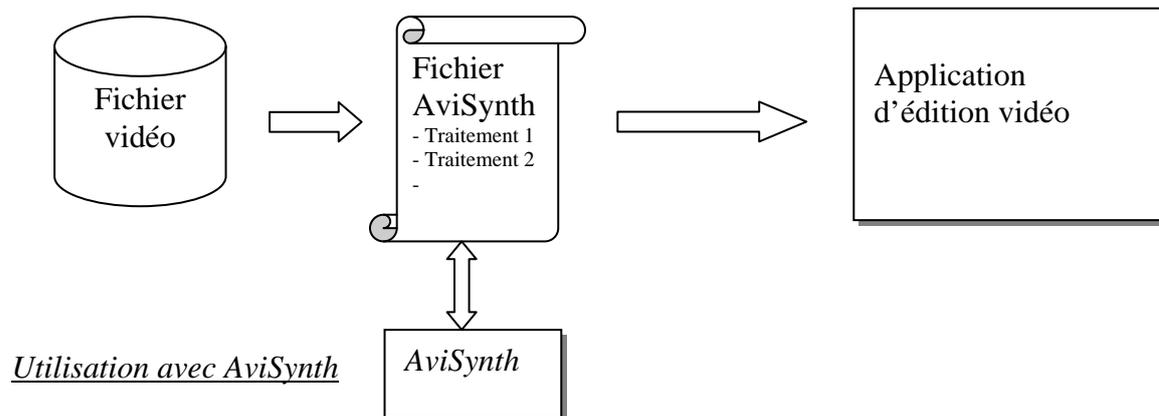
*AviSynth* permet de réaliser tout un tas de traitements vidéo sans nécessiter de volumineux fichiers intermédiaires. On peut ouvrir un fichier *AviSynth* comme on ouvrirait un fichier vidéo classique. Ce fichier ne contient pas les données vidéo mais les traitements que *AviSynth* doit effectuer sur un (ou plusieurs) fichier(s) vidéo avant de fournir le résultat à l'application qui a ouvert le fichier. De plus *AviSynth* peut utiliser directement les filtres de *VirtualDub*, on pourra donc utiliser le filtre de sous-titrage.

Dans une utilisation classique d'une application d'édition vidéo, on utilise la commande *Ouvrir* du menu *Fichier* pour sélectionner et charger un fichier vidéo. L'application affiche alors les images de la vidéo lues dans le fichier vidéo.



Utilisation classique d'un fichier vidéo

*AviSynth* s'intercale entre les deux, l'application ouvre le fichier *AviSynth*, le logiciel *AviSynth* démarre alors automatiquement pour lire le fichier vidéo source, effectuer les traitements spécifiés et fournir les images traitées à l'application de traitement vidéo



Un fichier *AviSynth* est en fait un simple fichier texte listant les opérations à effectuer. On peut le résumer de la manière suivante :

```

Chargement du fichier vidéo à traiter
  Traitement 1
  Traitement 2
  ...
Fourniture du résultat
  
```

Voici par exemple un script convertissant une vidéo en anaglyphe (filtrage simple) :

```

a = AviSource("mavideo.avi").ConvertToRGB()
l = a.Crop(0,0,Floor(a.width/2),a.height)
r = a.Crop(Floor(a.width/2),0,Floor(a.width/2),a.height)
return MergeRGB(l.ShowRed(), r.ShowGreen(), r.ShowBlue())
  
```

*AviSynth* va donc nous permettre de réaliser la solution 1 sans pour autant s'encombrer de fichiers vidéo intermédiaires. Le traitement *AviSynth* sera donc le suivant :

```

Chargement du fichier vidéo stéréoscopique à traiter
  Couper la vidéo pour obtenir les images gauche et droite
  Application des sous-titres sur la partie gauche
  Application des sous-titres sur la partie droite
  Recoller les parties gauche et droite en une image stéréoscopique
Retourner les images sous-titrées
  
```

Ce qui, dans le langage de script d'*AviSynth*, donne ceci :

```

# Charger la vidéo stéréoscopique et la convertir dans l'espace de
# couleur RVB (nécessaire pour utiliser les filtres de VirtualDub)
video = DirectShowSource("ElPinacate_720p_X.avi").ConvertToRGB32()

# Calculer la largeur d'une vue du couple
demilargeur = Round( video.width / 2 )

# Extraire les parties gauche et droite (ici en mode croisé)
droite = video.Crop(0, 0, demilargeur, 0)
gauche = video.Crop(demilargeur, 0, demilargeur, 0)

# Charger le filtre VirtualDub de sous-titrage et lui donner un nom
LoadVirtualDubPlugin("Subtitler.vdf", "Soustitreur", 1)

# Appliquer le sous-titrage sur les deux parties gauche et droite
gS = gauche.Soustitreur(1, "demo.ssa", 0, 0)
dS = droite.Soustitreur(1, "demo.ssa", 0, 0)

# Retourner les deux parties sous-titrées recollées ensemble
Return StackHorizontal(gS, dS)
  
```

Les lignes commençant par le signe dièse (#) sont des commentaires. Pour plus de clarté, les chemins d'accès aux fichiers externes (vidéo, filtre et sous-titres) ont été omis ; ils doivent bien sûr être précisés dans le vrai script qui sera utilisé.

Enregistrer ce script dans un fichier avec l'extension ".avs", par exemple *faireSoustitrage.avs*.  
Charger ensuite ce fichier dans *VirtualDub*.



*Cette fois nous obtenons bien un sous-titrage stéréoscopiquement correct !*

## Écriture du fichier de sous-titre

Jusqu'à maintenant nous avons simplement utilisé le fichier de démonstration du filtre de sous-titrage ce qui est d'un intérêt pratique limité. Par contre, c'est une bonne base pour écrire son propre fichier de sous-titres, un sous-titre ayant un bon nombre de paramètre : marges, couleurs, position, ... (on se référera à la documentation de référence pour en explorer toutes les subtilités !).

Le fichier de script est aussi un fichier texte mais avec l'extension ".ssa". Il comporte trois sections :

```
[Script Info]
; Informations et paramètres généraux du script

[V4 Styles]
; Styles des textes

[Events]
; Les lignes de sous-titres
```

Les lignes commençant par un point-virgule (;) sont des commentaires.

Exemple de définition de style :

```
Style: MainP,Arial,14,65535,65280,65535,0,0,0,1,2,4,1,12,12,4,0,0
```

Cette ligne définit un style de texte appelé « MainP » utilisant une police « Arial » de corps 14.  
Les paramètres qui suivent sont :

- Couleur primaire,
- Couleur secondaire,
- Couleur tertiaire,
- Couleur de fond,
- Gras (0: non, 1: oui),
- Italique (0: non, 1: oui),
- Style de bordure,
- Contour,
- Ombrage,
- Alignement,
- Marge gauche,
- Marge droite,
- Marge verticale,

- Transparence,
- Encodage.

Exemple de définition de sous-titre :

```
Dialogue: Marked=0,0:00:16.00,0:00:21.00,MainP,,0000,0000,0000,!Effect,Texte
```

Cette ligne définit un sous-titre de type « Dialogue ». Le texte « Texte » sera affiché du temps 16 secondes au temps 21 secondes (donc une durée de 5 secondes). Le style « MainP » est utilisé.

Les textes apparaissent en bas à gauche par défaut. On peut ajouter des commandes spéciales pour spécifier un autre alignement ou changer le format du texte. Voici les principales commandes :

- `{\a2}` : Texte centré et en bas,
- `{\a5}` : Texte en haut à gauche,
- `{\a3}` : Texte à droite et en bas,
- `{\b1}xxx{\r}` : mise en gras de « xxx »,
- `{\i1}xxx{\r}` : mise en italique de « xxx »,
- `{\fs20}xxx{\r}` : « xxx » affiché en taille 20,
- `{\fnCourier New}xxx{\r}` : « xxx » affiché avec la police *Courier*,
- `{\c&H0000FF&}xxx{\r}` : « xxx » affiché avec une couleur particulière (ici le rouge),
- `\N` : forcer un saut de ligne.



Sous-titre placé en haut pour ne pas masquer la flèche.

## Gestion de la profondeur

Prenons pour exemple cette image. Elle a été prise de trop près : l'enfant en bas à gauche et le grillage à droite sont en jaillissement (c'est une violation manifeste de la fenêtre stéréo !). Mais on souhaite toutefois l'inclure dans le diaporama malgré ce défaut. Le positionnement du sous-titre devient délicat : à gauche il masquerait les visages mais à droite il apparaît comme incrusté dans le grillage et devient difficile à lire.



Le texte est en effet affiché dans le plan de la fenêtre stéréoscopique. Ceci est dû au fait qu'on utilise le même fichier de sous-titres pour les images gauches et les images droites de la vidéo.

Voici la ligne correspondante dans le fichier de sous-titres :

```
Dialogue: Marked=0,0:04:02.00,0:04:09.00,MainP,,0000,0000,0000,!Effect,{\q1\a3}The city
also has an aquarium to get\Nacquainted with the local wildlife.
```

Notez la commande *a3* pour mettre le texte en bas à droite et le *\N* pour forcer un retour à la ligne.

Les trois paramètres **0000** sont utilisés pour spécifier les marges gauche, droite et verticale. On les laisse généralement à zéro car on utilise les marges définies au niveau des styles.

Pour positionner le texte en profondeur nous allons :

- Utiliser non pas un mais deux scripts de sous-titre,
- Modifier légèrement la marge de la ligne correspondante pour faire sortir (ou rentrer) le texte.

On duplique donc le fichier de sous-titre : *sousTitres.ssa* est copié en *sousTitres\_G.ssa* et *sousTitres\_D.ssa*. On modifie également le script *AviSynth* pour prendre en compte des fichiers de sous-titres différents pour la gauche et la droite :

```
# Appliquer le sous-titrage sur les deux parties gauche et droite
gS = gauche.Soustitreur(1, "sousTitres.ssa", 0, 0)
dS = droite.Soustitreur(1, "sousTitres.ssa", 0, 0)
```

devient :

```
# Appliquer le sous-titrage sur les deux parties gauche et droite
gS = gauche.Soustitreur(1, "sousTitres_G.ssa", 0, 0)
dS = droite.Soustitreur(1, "sousTitres_D.ssa", 0, 0)
```

On modifie le fichier de sous-titre droit pour modifier légèrement la marge droite (si le texte était aligné à gauche, on modifierait la marge gauche dans le fichier gauche).

La ligne devient (uniquement dans le fichier droit !) :

```
Dialogue: Marked=0,0:04:02.00,0:04:09.00,MainP,,0000,0016,0000,!Effect,{\q1\a3}The city
also has an aquarium to get\Nacquainted with the local wildlife.
```



*Le sous-titre apparait maintenant légèrement en avant du grillage.*

Note : On ne peut pas utiliser de valeurs négatives pour les marges. Si on souhaite faire « rentrer » le texte, c'est-à-dire le pousser derrière la fenêtre stéréoscopique, il faut jouer sur la même marge mais du côté opposé du couple. Par exemple, pour un texte aligné à gauche, on modifiera la marge gauche dans le fichier droit.



*Exemple de texte placé derrière la fenêtre stéréoscopique.*

## Utilisation de *Stereoscopic Player*

Le logiciel de lecture vidéo *Stereoscopic Player* permet de lire un grand nombre de formats vidéos stéréoscopiques et de les jouer en proposant un nombre tout aussi important de formats d'affichage (anaglyphe, entrelacé, côte-à-côte, double-projecteur, lunettes actives,...). Son seul défaut est donc de ne pas être gratuit (39 € pour une licence personnelle) mais la version de démonstration téléchargeable gratuitement permet de se faire une idée de l'intérêt du logiciel. Notez aussi qu'une version spécifique est fournie par *nVidia* avec son kit 3D Vision sous le nom de « *nVidia 3D Vision Video Player* ».

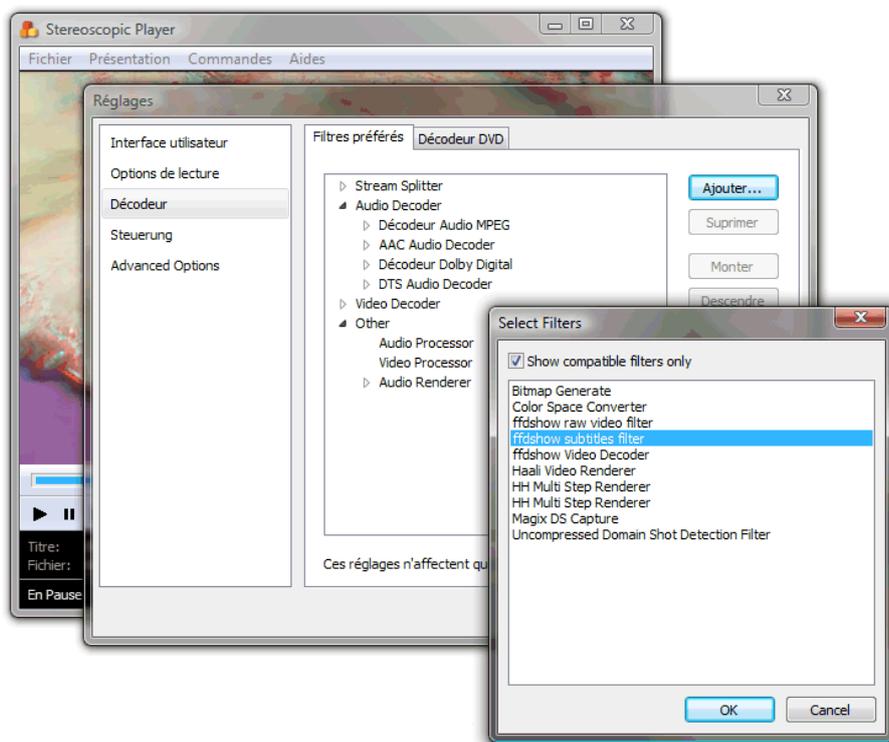
On pouvait aussi lui reprocher de ne pas gérer les sous-titres mais les gens du site *3D Vision Blog* ont trouvé la solution pour afficher des sous-titres sur une vidéo stéréoscopique. Il s'agit d'un sous-titrage par superposition, ajoutés au moment de la lecture et donc désactivables si besoin. Voire on peut avoir, pour une même vidéo, plusieurs fichiers de sous-titres dans des langues différentes. La solution repose aussi sur l'utilisation du multi-codec *ffmpeg* qu'il est toujours intéressant d'installer sur son ordinateur.

On supposera donc par la suite que *Stereoscopic Player* et *ffmpeg* sont installés et fonctionnent correctement.

## Configuration de *Stereoscopic Player*

Instructions traduite du site *3D Vision Blog*.

Lancer *Stereoscopic Player* et sélectionner la commande *Réglages...* du menu *Fichier*. Dans la fenêtre de configuration qui s'ouvre, sélectionner *Décodeur* dans la liste de gauche puis l'onglet *Filtres préférés*. Ouvrir la liste appelée *Other*, sélectionner *Video Processor* (tout n'a pas été traduit en français dans le logiciel !) et cliquer le bouton *Ajouter*. Dans la liste des filtres qui s'affichent dans une nouvelle fenêtre, sélectionner le filtre *ffmpeg subtitles filter* et cliquer le bouton *OK*.

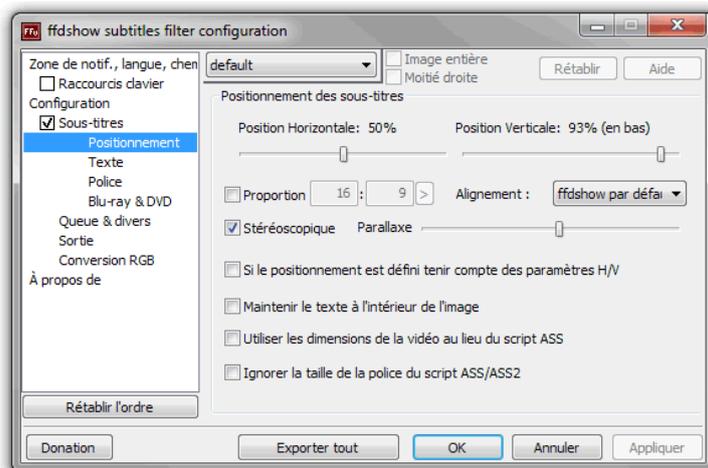
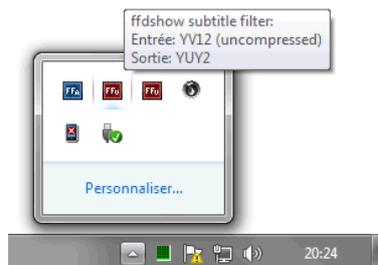


Rassurez-vous, cette opération n'est à effectuer qu'une seule fois ! Elle a pour but d'indiquer à *Stereoscopic Player* de charger le filtre de sous-titrage quand on charge une vidéo.

## Configuration de ffdshow

Ouvrons maintenant notre fichier vidéo dans *Stereoscopic Player* et mettons-le en pause dès que la lecture a démarré. Observons maintenant la zone des icônes de notification en bas à droite de l'écran et cherchons une petite icône carrée rouge avec le texte « FFV » écrit dessus.

Il peut y avoir plus d'une icône FFV si la vidéo est décodée par *ffdshow*, passer la souris dessus et double-cliquer celle dont l'info-bulle dit « *ffdshow subtitles filter* ». La fenêtre de configuration suivante doit alors apparaître.



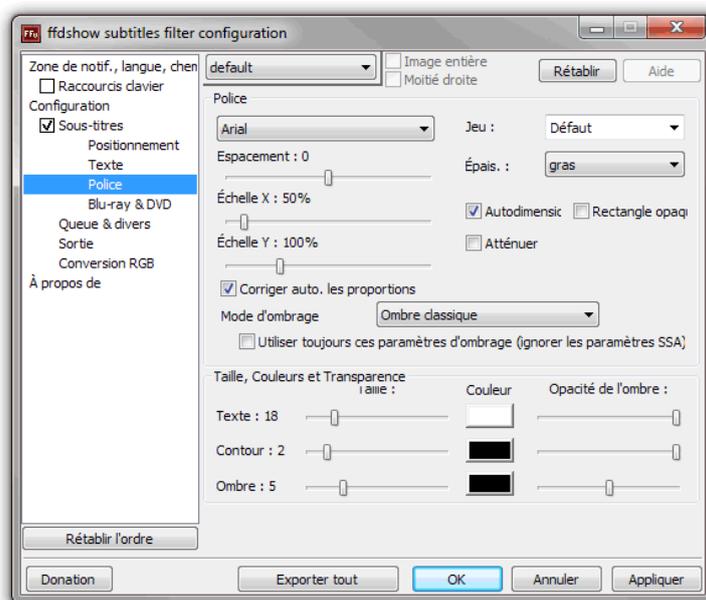
Dans cette fenêtre, cocher la case d'option *Sous-titres* à gauche pour que les sous-titres soient affichés puis sélectionner *Positionnement* et, dans le panneau de droite, cocher l'option *Stéréoscopique* pour que les sous-titres s'affichent sur les vues gauches et droite. Le curseur *Parallaxe* permet de placer les sous-titres en avant ou en arrière de l'écran.

## Affichage des sous-titres

Le plus simple est de nommer le fichier de sous-titre avec le même nom que le fichier vidéo et de les placer dans le même dossier. Le filtre *ffdshow* reconnaît les fichiers de sous-titres SSA et SRT (un format plus simple que le SSA). Après avoir vérifié que le fichier de sous-titres est bien présent, ouvrons à nouveau le fichier vidéo avec *Stereoscopic Player*, les sous-titres s'affichent !



Note : Si la vidéo est au format côte-à-côte compressé, les sous-titres vont apparaître très allongés. Ouvrir la fenêtre de configuration du filtre, sélectionner à gauche la ligne *Police* et dans le panneau de droite ajuster l'échelle horizontale *Échelle X* à 50%.



Il est aussi à noter qu'il n'est pas nécessaire de désactiver cette configuration des sous-titres pour jouer des vidéos n'en ayant pas besoin. Par défaut le filtre de sous-titrage de *ffdshow* cherche un fichier de sous-titres ayant le même nom que la vidéo à jouer. Si un tel fichier n'est pas trouvé, la vidéo sera jouée normalement et rien ne sera affiché en plus.

## Liens internet

Le sous-titrage dans *Wikipedia* : <http://fr.wikipedia.org/wiki/Sous-titrage>

*VirtualDub* : <http://www.virtualdub.org>

Tutoriels en français : <http://www.tutoriels-animes.com/tutoriels-pour-virtualdub.html>

D'autres tutoriels : <http://leblogaj.over-blog.com/article-21379957.html>

Utilisation 3D : <http://www.angelfire.com/film/puppetkite/VirtualDub3D>

Filtre *subtiter* : [http://www.virtualdub.org/virtualdub\\_filters.html](http://www.virtualdub.org/virtualdub_filters.html)

Filtre *StereoTools* : <http://www.stereoscopie.fr/tech/vdubfilters.php>

*AviSynth* : [http://avisynth.org/mediawiki/Main\\_Page/fr](http://avisynth.org/mediawiki/Main_Page/fr)

Des tutoriels pour *AviSynth* :

<http://neuron2.net/www.math.berkeley.edu/benrg/avisynth-tutorial.html>

<http://www.videohelp.com/tools/AviSynth>

<http://www.animemusicvideos.org/guides/avtech/avisyntha.html>

<http://www.animemusicvideos.org/guides/avtech/avspostqual.html>

Format de sous-titre *SSA* : [http://fr.wikipedia.org/wiki/SubStation\\_Alpha](http://fr.wikipedia.org/wiki/SubStation_Alpha)

Éditeur de sous-titres *Aegisub* : <http://www.aegisub.org>

*Stereoscopic Player* (39 €) : <http://3dtv.at/Products/Player>

Multi-codec *ffdshow* : <http://www.xvidvideo.ru/ffdshow-tryouts-project-x86-x64>

3D Vision Blog : <http://3dvision-blog.com>