RFC 9043 : FFV1 Video Coding Format Version 0, 1, and 3

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 24 août 2021

Date de publication du RFC : Août 2021

https://www.bortzmeyer.org/9043.html

Ce RFC décrit FFV1, un encodage de flux vidéo sans pertes, et sans piège connu, par exemple de brevet.

Certains encodages vidéo font perdre de l'information, c'est-à-dire que, après décompression, on ne retrouve pas tout le flux original. Ces encodages avec pertes tirent profit des limites de l'œil humain (on n'encode pas ce qui ne se voit pas) mais peut poser des problèmes dans certains cas. Par exemple, si on veut stocker une vidéo sur le long terme, il est certainement préférable qu'elle soit aussi « authentique » que possible, et donc encodée sans pertes. Les organisations qui font de l'archivage sur le long terme (comme l'INA) ont tout intérêt à privilégier un format sans pertes et libre, qui sera encore utilisable dans 50 ans, plutôt que de choisir les derniers gadgets à la mode. Si on souhaite faire analyser cette vidéo par autre chose qu'un œil humain, par exemple un programme qui souhaite avoir la totalité des informations, là encore, le « sans perte » est nécessaire. FFV1 est sans perte, pour le plus grand bénéfice de l'archivage et de la science. C'est le premier codec sans perte documenté par l'IETF.

FFV1 ne date pas d'aujourd'hui, et le RFC traite d'un coup trois versions de FFV1 (section 4.2.1) :

— La version 0 <https://git.videolan.org/?p=ffmpeg.git;a=commit;h=b548f2b91b701e1235608ac88
qui date de 2006,
— la version 1 <https://git.videolan.org/?p=ffmpeg.git;a=commit;h=68f8d33becbd73b4d0aa277f4
de 2009,
— la version 2 qui n'était qu'expérimentale et n'a pas été officiellement utilisée,
— la version 3 <https://git.videolan.org/?p=ffmpeg.git;a=commit;h=abe76b851c05eea8743f6c899
la dernière, en 2013.

Le nom FFV1 veut dire « FF Video 1 » et FF est une référence à FFmpeg, le programme de référence (qui a largement précédé la spécification officielle) dont le nom venait de "*Fast Forward*" (avance rapide).

Je ne vais pas vous décrire les algorithmes de FFV1, ça dépasse largement mes compétences. Comme le note le RFC, il faut d'abord connaitre le codage par intervalle, et le modèle YCbCr. Je vous renvoie donc à l'article en français d'un des auteurs <https://linuxfr.org/news/ffv1-un-format-video-sans-perte-
et bien sûr au RFC lui-même. FFV1 utilise souvent des techniques un peu anciennes, pour éviter les problèmes de brevet.

Le RFC utilise du pseudo-code pour décrire l'encodage FFV1. Ce pseudo-code particulier est assez proche de C. Vous verrez donc du pseudo-code du genre :

```
for (i = 0; i < quant_table_set_count; i++) {
  states_coded
  if (states_coded) {
      for (j = 0; j < context_count[ i ]; j++) {
          for (k = 0; k < CONTEXT_SIZE; k++) {
              initial_state_delta[ i ][ j ][ k ]
          }
      }
  }
}
```

Le type de média `video/FFV1` a été enregistré à l'IANA <https://www.iana.org/assignments/media-types/media-types.xhtml#video> pour les flux encodés en FFV1.

FFV1 décrit un encodage d'un flux vidéo, pas un format de conteneur. Le flux en question doit donc être inclus dans un conteneur, par exemple aux formats AVI, NUT ou Matroska (section 4.3.3 pour les détails de l'inclusion dans chaque format).

Comme pour tout codec utilisé sur le grand méchant Internet, un programme qui lit du FFV1 doit être paranoïaque et ne doit pas supposer que le flux vidéo est forcément correct. Même si le contenu du fichier est délibérement malveillant, le décodeur ne doit pas allouer de la mémoire à l'infini ou boucler sans fin (ou, pire encore, exécuter du code arbitraire par exemple parce qu'il y aura eu un débordement de pile ; FFV1 lui-même ne contient pas de code exécutable). Le RFC (section 6) donne l'exemple d'un calcul de taille d'une image où on multiplie la largeur par la hauteur, sans plus de précautions. Si cela provoque un dépassement d'entier, des tas de choses vilaines peuvent arriver. Un exemple de précaution : FFmpeg a été soumis à des flux FFV1 corrects et à des données aléatoires, tout en étant examiné par Valgrind et le vérificateur de Clang <https://clang.llvm.org/docs/AddressSanitizer.html> et aucun accès mémoire anormal n'a été détecté.

Question mises en œuvre, FFV1 est suffisamment ancien pour que de nombreux programmes sachent le décoder. L'implémentation de référence est FFmpeg (qui sait aussi encoder en FFV1 <https://trac.ffmpeg.org/wiki/Encode/FFV1>). Il y a également un décodeur en Go <https://github.com/dwbuiten/go-ffv1>, développé en même temps que la spécification et il y a aussi MediaConch <https://mediaarea.net/MediaConch> dont le développement a permis de détecter des incohérences entre le projet de spécification et certains programmes. La spécification elle-même a été développée sur GitHub <https://github.com/FFmpeg/FFV1> si vous voulez suivre son histoire, et son futur (une version 4 est prévue).

Comme exemple d'une vidéo encodée en FFV1, j'ai pris les 20 premières secondes de mon exposé au FOSDEM 2021 <https://fosdem.org/2021/schedule/event/retro_gemini/>. Le fichier (pour 20 secondes de vidéo !) fait 60 mégaoctets : la compression sans perte est évidemment moins efficace que si on accepte les pertes. Le fichier a été produit par FFmpeg (`ffmpeg -i retro_gemini.webm -vcodec ffv1 -level 3 -to 00:00:20 retro_gemini.mkv`, notez que, si FFV1 lui-même est sans perte, si vous encodez en FFV1 à partir d'un fichier déjà comprimé avec perte, vous ne recupérez évidemment pas ce qui a été perdu). mediainfo vous montre son contenu :

```
% mediainfo retro_gemini.mkv
General
...
Format                                   : Matroska
Movie name                               : Gemini, a modern protocol that looks retro
...
```

```
DATE                                     : 2021-02-07
EVENT                                    : FOSDEM 2021
SPEAKERS                                 : Stéphane Bortzmeyer

Video
ID                                       : 1
Format                                   : FFV1
Format version                           : Version 3.4
Codec ID                                 : V_MS/VFW/FOURCC / FFV1
Duration                                 : 20 s 0 ms
Bit rate mode                            : Variable
Bit rate                                 : 25.7 Mb/s
Width                                    : 1 280 pixels
Height                                   : 720 pixels
Frame rate mode                          : Constant
Frame rate                               : 25.000 FPS
Color space                              : YUV
Compression mode                         : Lossless
...
```

Ou bien avec ffprobe :

```
% fprobe -show_format -show_streams retro_gemini.mkv
...
  Input #0, matroska,webm, from 'retro_gemini.mkv':
  Metadata:
    title           : Gemini, a modern protocol that looks retro
    DATE            : 2021-02-07
...
[STREAM]
index=0
codec_name=ffv1
codec_long_name=FFmpeg video codec #1
codec_type=video
codec_time_base=1/25
codec_tag_string=FFV1
width=1280
height=720
...
```

Merci à Jérôme Martinez pour sa relecture attentive.