

### 3. FFmpeg を利用したメディアエンコーダの制作

貞友 拓海

#### 1. 研究概要

デジタルカメラで撮影した動画や、インターネット上から入手した動画などは、パソコン上ではそのまま再生できるが、iPod 等のポータブルデバイスへそのまま転送しても再生できないことが殆どである。これはポータブルデバイスが対応している動画の形式が非常に限られているからである。

これを解決するためには、動画をポータブルデバイスの対応している形式に変換する作業が必要になる（図1）。

今回、その変換作業が容易にできるソフトウェア、いわゆるエンコーダの作成を課題研究のテーマにした。



図1 動画の変換

#### 2. 研究の具体的内容

##### (1) 開発に使用するソフトウェア

エンコーダを一から開発するには非常に高度なプログラミング技術が必要になるため、FFmpeg と呼ばれるソフトウェアを利用する。

開発ソフトウェアには、Microsoft 社の Visual Studio 2008 を使用し、C#言語で開発した。

##### (2) FFmpeg について

FFmpeg とは、動画と音声を変換することのできるフリーでオープンソースのソフトウェアである。オープンソースとは、ソフトウェアの設計図であるソースコードを公開して、誰でも自由に改変、配布ができるようにすることである。

対応している形式が多く（表1）、オープンソースであるため、スマートフォンやブルーレイプレーヤなどのデバイスにも組み込まれている。

表1 FFmpeg が変換に対応している代表的な動画・音声の形式

ASF	AVI	Matroska
FLV	WAV	QuickTime
3GPP	MPEG-1	MPEG-2
MPEG-4	Ogg	その他多数

##### (3) C#言語、Visual Studio について

C#は、2000年にMicrosoft社が発表したオブジェクト指向プログラミング言語である。C言語やC++言語をベースに改良し、生産性、機能を向上させた言語になっている。Visual Studioは、Microsoft社が開発している統合開発環境である。フォームにボタンやテキストボックスなどを貼り付け、容易にアプリケーションを開発することができる（図2）。

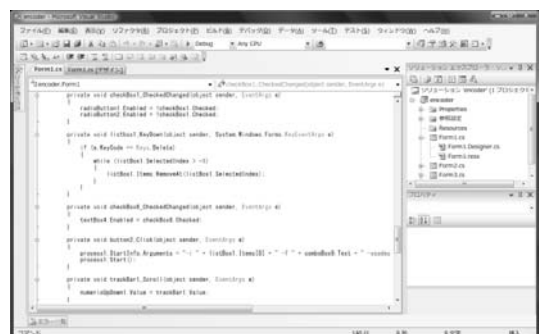


図2 Visual Studio 2008 での開発の様子

#### (4) GUI、CUI について

FFmpeg は CUI のソフトウェアである。CUI とは、キャラクターユーザインタフェース (Character User Interface) の略で、すべての操作をキーボードで行い、文字によって情報を表示するユーザインタフェースのことである (図 3)。すべての操作が文字によって行われるため、PC 初心者には分かりづらい。

例えば、FFmpeg に動画を変換させるには、次のようなコマンドをキーボードから入力する必要がある。

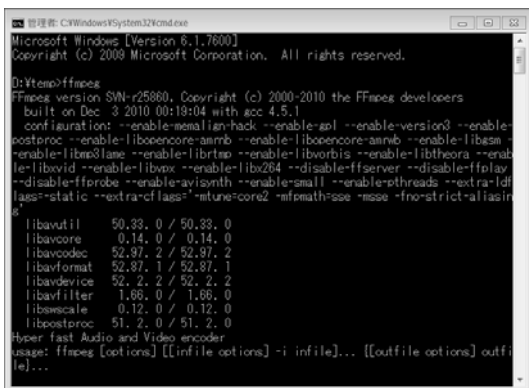
動画を MPEG-4/AVC 形式に変換する

コマンドライン入力例(ビットレート固定)

```
ffmpeg -i inputfile -f mp4 -vcodec libx264
-acodec libfaac -vb 256k -ab 64k
outputfile.mp4
```

一方、GUI はグラフィカルユーザインタフェース (Graphical User Interface) の略で、ウィンドウ、ボタン、アイコン等のグラフィックス、マウス等のポインティングデバイスを用いて、直感的な操作ができるユーザインタフェースのことである。私たちが普段利用しているソフトウェアの殆どがこれにあたる。

FFmpeg は単体では使いにくいので、GUI の操作を可能にする、GUI ラッパーと呼ばれる種類のソフトウェアの開発をした (図 4)。



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

D:\temp>ffmpeg
FFmpeg version SVN-r25860, Copyright (c) 2000-2010 the FFmpeg developers
built on Dec  3 2010 00:19:04 with gcc 4.5.1
configuration: --enable-memalign-hack --enable-zlib --enable-version3 --enable-
postproc --enable-libopencore-aac --enable-libopencore-aac --enable-libas
--enable-libx264 --enable-libx264 --enable-libx264 --enable-libx264 --enable-
libvid --enable-libx264 --enable-libx264 --enable-libx264 --enable-libx264 --enable-
--enable-libx264 --enable-libx264 --enable-libx264 --enable-libx264 --enable-
--enable-libx264 --enable-libx264 --enable-libx264 --enable-libx264 --enable-
--enable-libx264 --enable-libx264 --enable-libx264 --enable-libx264 --enable-
libavutil 50.33.0 / 50.33.0
libavcore  0.14.0 / 0.14.0
libavcodec 52.97.2 / 52.97.2
libavformat 52.87.1 / 52.87.1
libavdevice 52. 2. 2 / 52. 2. 2
libavfilter 1.86. 0 / 1.86. 0
libswscale  0.12. 0 / 0.12. 0
libpostproc 51. 2. 0 / 51. 2. 0
Hyper fast Audio and Video encoder.
usage: ffmpeg [options] [[infile options] -i infile]... [[outfile options] outfi
le]...
```

図 3 コマンドプロンプトから実行した FFmpeg (CUI)

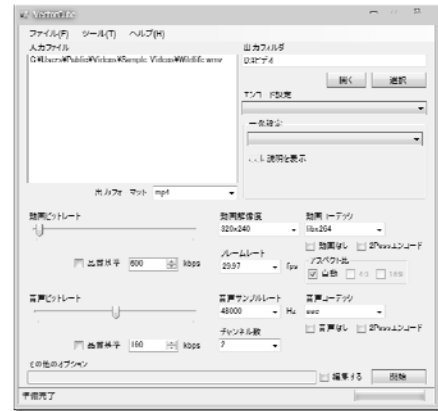


図 4 完成したソフトウェア

### 3. 研究のまとめ

この研究は、自分の手で使いやすいエンコーダを作ること、C#言語でのプログラミング技術の向上や、動画・音声の形式の知識の習得を目指して始めた。

C#の経験は、授業で少し学習したくらいだったので、インターネットで調べ、簡単なプログラムを作ったりして知識を得た。

動画や音声の形式は、非常に多種多様で、同じ形式の中でもさらに細かく仕様が分かれているので、どのようなものなのかを把握するのにとても苦労した。

まだ、C#や動画、音声の仕様を完全には理解できていなかったり、未実装の機能や、バグがあったりするので、これからも開発を続けてさらに完成度を高めていきたい。

### 参考文献

- FFmpeg  
<http://www.ffmpeg.org/>
- FFmpeg - Wikipedia  
<http://ja.wikipedia.org/wiki/FFmpeg>
- 最新 FFmpeg - MobileHackerz Knowledge base Wiki  
<http://mobilehackerz.jp/archive/wiki/index.php?%BA%C7%BF%B7FFmpeg>
- 緑のバイク ★ 初めての C#  
[http://homepage3.nifty.com/midori\\_no\\_bike/CS/index.html](http://homepage3.nifty.com/midori_no_bike/CS/index.html)