

一种基于批处理的服务器端视频转换模块的实现^①

赵鑫莹

(山东师范大学 传播学院, 济南 250014)

摘要: 实现了一种将当前几乎所有格式的视频文件定时、自动、批量的处理成可在 Web 页面播放的 flv 视频的模块。首先介绍了当前主流视频格式, 以及 Mencoder 和 Ffmpeg 组件; 然后介绍该视频转换模块的工作流程设计; 最后介绍其主要子模块的功能实现, 以及在实际应用中应注意的问题。

关键词: 批处理命令; 视频转换; 定时处理; 自动检测

Realization of a Video Conversion Module Based on Batch Programs

ZHAO Xin-Ying

(Communication College, Shandong Normal University, Jinan 250014, China)

Abstract: This paper realizes a module which can convert almost all formats of current video files to files of flv format which can be played on the Web pages timely, automatically and batchingly. Firstly, it introduces current main video formats, along with Mencoder and Ffmpeg components. Secondly, it introduces the workflow design of this module. Finally, the paper introduces function realization of its submodules, and matters we should pay attention to in practical applications.

Keywords: batch program; video conversion; timed process; automatic detection

在为某银行开发“员工之家”视频点播系统时, 遇到这样的问题: 一方面, 该单位购买了一批视频光盘, 需要加入视频资源库; 另一方面, 需要确保员工能够从 web 客户端上传视频, 并且要求可上传的视频格式尽可能多; 最后, 所有的视频必须能够通过浏览器在网页上直接播放。

考虑到 flash 播放器插件的在浏览器上的普及性 (Adobe 官方称 90% 以上), 决定将所有视频转换成可以通过 FlashPlayer 播放的 flv 格式视频文件。为了确保转换的效率和稳定性, 决定放弃传统的由客户端触发服务器端视频转换程序进行在线转换的模式, 而在服务器端运行一个批处理程序实现对视频文件的定时处理和自动检测。这样一方面可以避免在访问高峰期由于客户端同时调用服务器上多个视频处理程序, 而给服务器造成负载; 另一方面可以防止用户因网络问题或不当操作, 而造成的上传触发失败或多次重复触发。

1 主流视频格式及 Mencoder 和 Ffmpeg 组件

1.1 当前流行的视频格式

常见的视频格式有: MPEG-1、MPEG-2、MPEG-4、AVI、MOV、ASF、WMV、nAVI、QuickTime、DV-AVI、RM、RMVB、FLV 等, 其扩展名有 .mov、.avi、.dat、.mpg、.mpeg、.vob、.asf、.mpe、.wmv、.asx、.mkv、.rmvb、.rm、.flv 等^[1]。

其中 VCD 光盘中的视频一般为 .dat 文件, DVD 光盘中视频文件一般为 .vob 文件。FLV 是一种流媒体格式。它的文件极小、加载速度极快, 可以使用 FlashPlayer 播放, 使得其成为网络播放视频的首选, 被广泛应用到优酷、土豆等大型视频网站。

1.2 Mencoder 和 Ffmpeg 组件

Ffmpeg 和 Mencoder 都是开源的用于视频转换的命令行工具, 被广泛应用于大多数的视频转换软件, 例如: 暴风影音播放器利用了 Ffmpeg, Mplayer 播放器利用了 Mencoder。本模块中, 我们借用 Ffmpeg 和

^① 收稿时间:2010-07-12;收到修改稿时间:2010-08-07

Mencoder 完成视频转换的底层操作。Mencoder 需要 drv43260.dll 和 libiconv.dll 组件, Ffmpeg 需要 pthreadGC2.dll 组件。本系统要求所有组件与批处理程序存于同一目录下。

Ffmpeg 的优点在于编码速度快, Mencoder 的优点在于可转换的视频种类多、质量高, 但是转换速度较 Ffmpeg 慢。例如: 扩展名为 .rm、.rmvb 的文件只能使用 Mencoder 转换; Ffmpeg 转换 wmv 和 asf 时经常出现花屏现象, 转换 .mkv 格式的效果也较差。

考虑到以上因素, 提出以下转换方案, 如表 1:

表 1 用于视频转换的视频格式集合分类

转换组件	可转换的视频类型
Ffmpeg	*.mov *.avi *.mpg *.mpeg *.mpe *.3gp
Mencoder	*.rm *.rmvb *.wmv *.asf *.asx *.mkv

2 视频转换模块工作流程设计

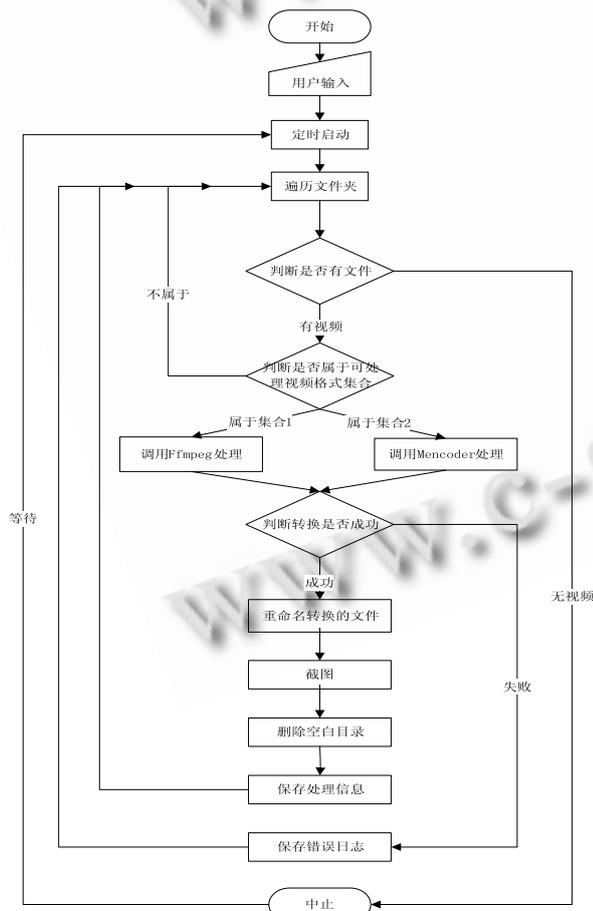


图 1 视频转换模块工作流程图
系统启动后, 显示用户参数输入界面。完成输

入后, 经过定时计时阶段, 主程序启动, 遍历源视频文件存储目录及其子目录。如果发现视频文件, 则判断其格式是否属于可转换的视频格式集合; 如果属于, 则调用相应的视频转换子模块, 执行视频转换。如果转换成功, 则执行文件重命名、截图、删除空目录、保存信息等操作; 一旦失败, 将错误信息写入错误日志。当遍历完源视频存储目录而没有发现视频文件时, 执行等待指令, 直到等足用户设定的启动时间, 再次触发主程序。模块工作流程图如图 1 所示。

3 主要子模块的功能实现

3.1 用户输入子模块

该模块为用户提供了参数输入接口。为普通用户提供了基本的输入参数项, 并提供用于视频转换的质量优先值、速度优先值、兼顾质量与速度的默认值三种快捷选项; 为高级用户除了提供基本输入参数外, 还提供用于视频转换的高级参数设置。基本用户输入参数如表 2 所示:

表 2 基本用户输入参数

名称	含义	取值
StartTime	定时时长 (单位分钟)	默认为 10
UserCategory	用户种类	0 为普通用户 1 为高级用户
SourceFilePath	源视频文件的存放路径	默认为 D 盘根目录
FileStorePath	处理后的文件保存路径	默认为 E 盘根目录
IsDelSourceFile	是否删除源文件	默认为删除
CatchPicWidth	截图的宽度	默认为空, 表示按原图宽度
CatchPicHeight	截图的高度	默认为空, 表示按原图高度
InfoPath	转换信息保存路径	默认为 F 盘根目录
ErrorPath	错误日志保存路径	默认为 F 盘根目录

高级用户输入参数如表 3 所示:

表 3 高级用户输入参数

名称	含义	默认值	速度 优先	质量 优先
VideoWidth VideoHeight	转换后视频画面的宽和高	不设置	不设置	不设置
AcValue	音频声道数 【Ffmpeg 参数】	2	2	2
AbValue	音频流量 【Ffmpeg 参数】	56	56	112
ArValue	输出采样率 【Ffmpeg 参数】	22050	11025	22050
BValue	视频流量 【Ffmpeg 参数】	200	200	400
RValue	每秒输出的帧数 【Ffmpeg 参数】	25	12	40

MbdValue	宏块决策算法 【Mencoder 参数】	0	1	2
CmpValue	像素运动比较函数 【Mencoder 参数】	0	0	6
VbStrateValue	I/P/B 帧策略 【Mencoder 参数】	0	1	2

3.2 定时触发子模块

该模块调用用户输入的定时时长(单位为: 分钟), 记录下触发定时时刻的时间, 与当前的时间进行比较, 当时间间隔等于定时时长时, 启动视频转换程序。

```

Rem 保存设置定时触发时的时间
Set /p StartTime= 请输入定时触发的时间,单位:分钟

Set /a f=0
Set tmpProcessFile=%tmp%\TmpRecorder.vbs
Set tmpTimeFile=%tmp%\TmpRecorder.lzy
Goto recordnow
:RecordNow
    Echo wscript.echo hour(time) ^&":"^&
minute(time) ^&":"^& second(time) > %tmpClockFile%
    cscript /nologo %tmpClockFile% >
%tmpTimeFile%
    Goto loop
:Loop
Rem 提取设置定时触发时的时间与当前时间相比
较
Set /a f=%f%+1
For /f "tokens=1-3 delims=:" %%a in
(%tmpTimeFile%) do (
Set /a _hr=%%a
Set /a _mr=%%b
Set /a _sr=%%c )
For /f "tokens=1-3 delims=:" %%A in ("%time%")
do (
Set /a hn=%%A
Set /a mn=%%B
Set /a sn=%%C )
Set /a hc=%hn%-_%_hr%
Set /a mc=%mn%-_%_mr%
Set /a mm=%hc%*60
Set /a mm=%mm%+%mc%
Rem 如果相差的分钟数正好与输入的定时时间
(单位为分钟) 相同, 则触发视频转换程序
If "%sn%"=="%_sr%" (
If "%mm%"=="StartTime%" goto start )
Goto Loop
Rem 开始执行视频转换程序
:Start
    
```

3.3 检测与判断子模块

该模块调用用户输入的源视频文件的存储目录, 检测该目录及其子目录下是否有视频文件。如果有, 则根据其所属的可转换视频集合(如表 1 所示)调用相应的视频转换子模块。

Rem 如果属于用 Ffmpeg 转换的视频集合, 调用 FfmpegConvert

```

For /R [SourceFilePath] %%I in (*.mov *.avi
*.mpg *.mpeg *.mpe *.3gp) do call :FfmpegConvert %I
%%~I %%~nI
    
```

Rem 如果属于用 Mencoder 转换的视频集合, 调用 MencoderConvert

```

For /R [SourceFilePath] %%I in(*.rm *.rmvb
*.wmv *.asf *.asx *.mkv) do call :MencoderConvert %I
%%~I %%~nI
    
```

3.4 视频转换子模块

该模块调用用户输入的与视频转换相关的参数, 利用 Ffmpeg 或 Mencoder 将视频文件处理成 flv 格式; 为防止文件重名, 对转换后的文件重命名。

3.4.1 使用 Ffmpeg 转换

该模块调用用户输入的参数, 如: 源文件路径、存储路径、音频和视频流量及画面大小等, 利用 Ffmpeg 执行视频转换^[2]。

```

Ffmpeg.exe -i [infile] -y -ab [AbValue] -ar
[ArValue] -ac [AcValue]2 -b [BValue] -qscale 9 -r
[RValue] -s [VideoWidth:VideoHeight] [outfile]
    
```

相关参数的详细说明及设置见表 2。

3.4.2 使用 Mencoder 转换

该模块调用用户输入的参数, 如: 源文件路径、存储路径、音视频取样的算法及画面大小等, 利用 Mencoder 执行视频转换^[3]。

```

Mencoder.exe [infile] -o [outfile] -of lavf -oac
mp3lame -lameopts abr:br=[BrValue] -ovc lavc -
lavcopts vcodec=flv:vbitrate=[VbitrateValue]
:mdb=[MbdValue]<0-2>:mv0:trell:v4mv:cbp:last_pred=[
LastpredValue]<0-99>:dia=[DiaValue]<-99-6>:cmp=[Cm
pValue]<0-2000>:vb_strategy=[VbStrategyValue]<0-2>
-vf scale=[VideoWidth:VideoHeight] -ofps [OfpsValue]
-srate [StrateValue]
    
```

有关设置项说明:

- of lavf 使用 libavformat 流合并器编码;
- oac mp3lame 使用 LAME 音频编解码器编码;
- ovc lavc 使用 libavcodec 编解码器编码?3?。

相关参数的详细说明及设置见表 3。

3.4.3 修改文件名

该模块将转换完成的视频文件以系统的日期和时

间作为其文件名, 来确保文件名的唯一性, 防止因文件重名而产生的文件替换。

```
Rem 获取文件扩展名
Set extension=%~x1
Rem 获取日期, 格式: 年月日
For /f "tokens=1-3 delims=-/ " %%A in ('date/T') do
set date=%%A%%B%%C
Rem 获取时间, 格式: 时分秒
For /f "tokens=1-3 delims=: " %%A in ("%time%")
do set time=%%A%%B%%C
Rem 修改文件名
Ren %1 %date%%time%%extension%
```

3.5 截图子模块

该模块调用用户输入的参数, 如: 图片抓取时间, 图片的宽和高, 利用 Ffmpeg 对转换出的 flv 文件执行图片抓取工作。

```
Rem 执行对视频截图, 可设置图片的宽度和高
Ffmpeg.exe -i [infile] -y -f image2 -ss 8 -sameq -t
CatchPicTime -s [CatchPicWidth]*[ CatchPicHeight]
[outpic]
```

3.6 空目录删除子模块^[4,5]

该模块遍历源文件的存储目录, 将空文件夹删除, 以利于对源文件的管理。

```
Rem 遍历当前目录及其子目录
For /f "delims=" %%i in (dir /ad /b /s) do (
Rem 把空目录的文件名写入 txt 文件
Dir /b "%i" | findstr .>nul || echo %%i
>>listnull.txt)
Rem 逐行读取 txt 文件中的空目录路径, 删除之
For /f %%i in (listnull.txt) do (
Rd /q %%i)
Rem 删除该 txt 文件
Del /q listnull.txt
```

3.7 信息存储子模块

该模块调用用户输入的保存处理信息的目录, 将视频处理过程中的信息如: 源视频的文件名, 转换后的视频的文件名, 处理的日期, 是否删除了源文件, 视频的时长等写入 xml 文件, 并保存到该目录。

```
Rem 创建保存信息的 xml 文档
Cd. >% InfoPath %\recorder.xml
Rem 建立 xml 文档的根节点
Echo ^<recorder^>^</recorder^> >>% InfoPath
%\recorder.xml
:InfoRecorder
Rem 建立名为 recorder_new 的 xml 文件
Cd. >% InfoPath %\recorder_new.xml
```

Rem 读取 recorder.xml 的内容并将</recorder>替换为相关信息

```
For /f %%a in (% InfoPath %\recorder.xml) do (
Set str=%%a
Set
str=!str:^<^/recorder^>=^<file^>^<FileName^>%
FileName %
^<^/FileName^>^<ConvertTime^>% ConvertTime
%
^<^/ConvertTime^>^<PrimaryName^>%
PrimaryName %
^</PrimaryName^>^<FileLength^>% FileLength %
^</FileLength^>^<IsDelSourceFile^>%
IsDelSourceFile %
^</IsDelSourceFile^>^</file^>^</recorder^>!
Echo !str! >> % InfoPath %\recorder_new.xml)
Rem 删除 recorder.xml
Del /q % InfoPath %\recorder.xml
Rem 将 recorder_new.xml 更名为 recorder.xml
Ren % InfoPath %\recorder_new.xml % InfoPath
%\recorder.xml
```

4 实际应用

该模块作为视频点播系统的视频转换子系统, 可以根据用户输入的参数定时触发、自动检测服务器的视频文件, 将其批量转换为可以直接在 web 页面播放的 flv 格式的视频, 并将处理信息写入 xml 文档, 可以作为 web 应用程序提取视频信息的数据库。本系统已在该单位服务器安全运行一年有余, 效果良好。需要注意的是, 对于内容的审核等问题还需要人工操作。

参考文献

- 1 武鹏. 常见视频格式的定义及视频压缩技术探讨. 中国科技信息, 2008, (2): 92-95.
- 2 FFmpeg Documentation. [2010-5-7]. <http://www.ffmpeg.org/documentation.html>
- 3 MPlayer 参考文档. [2010-5-10]. http://www.mplayer.org.cn/help_zh.html
- 4 邹县芳, 胡昆鹏. 批处理在提高 windows 管理效率中的神奇应用. 北京: 中国铁道出版社, 2008. 15-53.
- 5 Stanek WR. 王景新, 等译. Windows 命令行详解手册. 北京: 人民邮电出版社, 2009. 21-46.