

使用 FFMPEG 从视频文件提取图像

如果你需要从视频文件中提取图像，你可以从视频中提取所有的帧为图像文件。下面的命令行从 video.webm 提取所有的帧：

```
bash ffmpeg -i video.webm thumb
```

- -i 接输入视频文件
- thumb%04d.jpg 是输出文件。在这个例子中为 jpg 文件，因此 ffmpeg 将输出 jpg 文件。我们制定 thumb%04d.jpg 和 "%04d" 在文件名称中，FFMPEG 将提取图像按照 thumb0000.jpgthumb0001.jpg 等等。我们制定 ffmpeg 输出文件有 4 维"%04d"，我们可以改变它产生更多，比如"%6d"
- -hidden_banner 隐藏 ffmpeg 的编译信息。

在这个例子中仅仅使用 jpg 文件作为输出，但是我們也能使用像 png 或者 bmp 格式。

警告：如果我们从一个长视频获取所有的帧，我们可能获得成百上千张图片，磁盘容量可能是一个大问题。

1 仅仅提取一帧

因为提取所有的图像通常不太好处理，通常使用更多的是提取单个视频图像。这样能更好的从视频获取重要部分。

对于同样的 video.webm 文件，我们想提取一个图像的第 7 秒的图像，作为视频中我们认为重要的一部分。我们可以使用 ffmpeg 使用下面的目录提取这一时刻的帧：

```
ffmpeg -i video.webm -ss 00:00:07.000 -vframes 1 thumb.jpg
```

- -ss 我们使用这个参数给定时间 (00:00:07.000 是视频中第 7s)。这参数使得 ffmpeg 搜索这一秒然后处理，因此它提取这一瞬间的帧
- -vframes 这个参数告诉 ffmpeg 提取的帧数 (这里是一帧)。ffmpeg 将仅仅提取一张图片输出为 thumb.jpg。这里我们没有使用%04d 这样的参数，但是可以使用。

有时候我们检查图片的质量才发现他们质量和我们想要的不同 (从视频文件中提取帧可能导致低质量的帧)。一个好办法是从视频的重要部分提取超过一帧。我们可以使用下面的命令：

```
bash ffmpeg -i video.webm -ss 00:00:07.000 -vframes 3 thumb
```

这里我们使用%04d 输出，按照我们的要求输出 3 帧，因此我们得到 3 张 jpg 图像。结合这个命令我们将从提示中获得前三帧，之后 ffmpeg 搜索视频的第二个 7 秒。

2 在规定时间内提取帧

另一个用于从视频文件提取图像的方法是每 x 秒提取一张图片。用这个方法我们不会想第一个例子一样得到很多图片，我们不限从视频的特定时间选择图像。

ffmpeg -i video.webm -vf fps=1 thumb%04d.jpg -hide_banner 在这个例子中我们用一个新的参数：

-vf: 这个参数用于 ffmpeg 过滤视频。我们可以用一个过滤器。这个例子中我用”fps=1”, 因此 ffmpeg 将过滤视频提取一张图片（每秒一帧）输出。我们使用 thumb%04d.jpg 作为输出提取多个文件。

因此我们得到视频文件每秒的图像，thumb0000.jpg,thumb0001.jpg 等等。

我们已经使用这个过滤器每秒提取一张图片，但是我们可以使用它结合其他的参数提取图像。我们可以每 5 秒提取一张图像：

```
ffmpeg -i video.webm -vf fps=1/5 thumb%04d.jpg -hide_banner
```

在这个例子中我们使用 “fps=1/5” 指定 ffmpeg 将每 5 秒提取一张图片。我们可以改变这个参数为”fps=1/10” 每 10s 提取一张图像。

3 从视频中提取关键帧

关键帧（有时候也叫索引帧）是用作视频文件参考的图片。关键帧有点复杂，但是正如之前总结的他们可以用作索引以至于视频播放器能容易的搜索它，他们可以用做下面帧的参考以至于视频解码器存储关键帧和下面帧的不同而不是所有的帧。

这也是为什么我们能在这个帧上获得更好的图像的原因。视频中关键帧之间的时间周期。关键帧配置 当我们创建视频文件时可以被改变因此他也许有多种配置以至于作者习惯创建视频。周期通常是 2 到 5 秒。

为了从视频中提取关键帧，我们使用下面的命令：

```
ffmpeg -i video.webm -vf "select=eq(pict_type,I)" -vsync vfr thumb%04d.jpg -hide_banner
```

- -vf 我们用一个视频过滤器，这里的视频过滤器比之前的有点复杂。我们使用”select=eq(pic_typeI)” 将使得过滤器选择关键帧的所有图片（”pic_typeI” 查找图像类型索引，”eq” 查找等于，因此我们可以将它解释为选择索引图像的所有）
- -vsync vfr: 这个参数告诉过滤器使用一个变化的比特率视频同步。如果我们不用这个参数 ffmpeg 将不能找到唯一的关键帧，显示提取其它的帧可能不被正确处理。

在这个例子中我们从 14s 的视频中提取 6 张图像（因此视频每两秒有一个关键帧）但是我们之前已经说过，变量一带与视频文件。指令：1. 从某一个时间段获取数据

```
ffmpeg -ss 00:08:14 -t 00:10:31 -i Infernal.Affairs.mkv -vf fps=1 test%04d.jpg -hide_banner
```

4 ffmpeg 官方文档

4.1 语法

```
ffmpeg [global_options] {[input_file_options] -i input_url} ... {[output_file_options] output_
```

4.2 描述

ffmpeg 是一个非常快的从实时音频视频源频转化工具。它可以用高质量的多想滤波器在任意采样率和 resize 视频。ffmpeg 能通过指定-i 读取任意数量的文件（可以使正常文件，pipe，网络流，抓取设

备等等), 写任意数量的输出文件, 通过输出的 url 指定。命令行中任何将不能解释为一个选项考虑作为一个输出 url。

每个输入或者输出 url 能在原则上包含一些不同类型的流 (video/audio/subtitle/attachment/data)。允许数量和流的类型通过容器的格式限制。从输入选择流将自动进入输出使用-map 选项 (查看流选择章节)。

为了查询在选项中的输入文件, 你必须使用他们的索引 (0-based). 例如第一个输入文件是 0, 第二个为 1 等等。类似的, 流结合一个文件通过索引访问。例如 2:3 访问 4 个输入流的第三个输入文件。也可以查看流指定章节。

正如一个通常的规则用于下一个指定文件的选项。因此顺序很重要, 你可以在命令行中保存同样的选项多次, 每个发生然后被应用在下一个输入或者输出文件。这个规则是全局选项, 应该首先被指定。

不要混淆输入和输出文件, 首先制定所有的输入文件, 然后所有的输出文件。不要混合不同文件的选项。所有的选项仅仅用于狭义和输入或者输出文件和两个文件之间的重置。

- 设置输出视频的比特率为 64 kbit/s `ffmpeg -i input.avi -b:v 64 -bufsize 64k output.avi`
- 强制输出比特率为 24fps `ffmpeg -i input.avi -r 24 output.avi`
- 强制输入文件的比特率 (仅仅对于 raw 格式) 为 1fps 输出帧率 24fps: `ffmpeg -r 1 -i input.m2v -r 24 output`

选项也许需要原始的输入文件。