



EBook Gratuito

APPENDIMENTO

video

Free unaffiliated eBook created from
Stack Overflow contributors.

#video

Sommario

Di.....	1
Capitolo 1: Iniziare con il video	2
Osservazioni.....	2
Examples.....	2
Comprensione dei file multimediali stand-alone.....	2
Generale	3
video	3
Audio	5
Riepilogo delle analisi	5
Di Più	6
Capitolo 2: Capire una presentazione multimediale	7
Osservazioni.....	7
Examples.....	7
Comprensione dei file multimediali stand-alone.....	7
Generale	8
video	8
Audio	10
Riepilogo delle analisi	10
Di Più	11
Comprensione dei componenti delle tracce video.....	11
Comprensione delle presentazioni di streaming adattive DASH.....	14
Capitolo 3: Proporzioni video	18
Osservazioni.....	18
Examples.....	18
Visualizza proporzioni (DAR).....	18
Formato immagine (PAR).....	18
Rapporto di aspetto del campione (SAR).....	19
Proporzioni pixel.....	20
Titoli di coda	21

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [video](#)

It is an unofficial and free video ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official video.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capitolo 1: Iniziare con il video

Osservazioni

La riproduzione video è frontale e centrale in una vasta gamma di soluzioni moderne, con il software e gli standard in rapida evoluzione. Per capire questo campo, devi prima capire i molteplici aspetti coinvolti nel lavorare con il video:

- Le informazioni di colore grezzo acquisite dal mondo fisico sono comunemente *codificate* utilizzando un codec, un algoritmo il cui scopo è quello di rappresentare questi dati in una forma compressa, spesso sacrificando alcuni dettagli visivi in favore di una maggiore compressione.
- Per la riproduzione, viene eseguito l'algoritmo inverso - i dati vengono *decodificati* per diventare nuovamente informazioni di colore grezzo che possono essere fornite a un dispositivo di output (ad esempio un monitor).
- Tra codifica e decodifica, i dati compressi vengono *impacchettati* per l'archiviazione, il che può implicare la combinazione di tracce di tipi diversi in un singolo file o la segmentazione del contenuto in un numero elevato di segmenti piccoli.
- Il video viene *consegnato* al dispositivo dell'utente finale utilizzando una tecnologia di consegna, che può essere semplice come un download di file via HTTP o molto più complesso, che coinvolge feedback in diretta dall'infrastruttura di rete e adattamento automatico dei livelli di qualità.
- Il contenuto premium viene solitamente *crittografato* prima dell'imballaggio e può essere riprodotto solo su un lettore dotato di tecnologia DRM che garantisce la sicurezza della chiave di decodifica durante l'uso e protegge attivamente dall'acquisizione dell'output.

Mentre la porzione visiva è ovviamente dominante nella visibilità, l'audio e il testo svolgono anche un ruolo chiave nelle presentazioni multimediali, fornendo funzionalità multilingua che rendono i contenuti accessibili a un vasto pubblico. Nella maggior parte dei flussi di lavoro, le tracce audio e di testo vengono gestite in modo equivalente alle tracce video, vengono codificate, decodificate, pacchettizzate e consegnate lungo le stesse linee.

Tutti questi aspetti - e altro - devono essere curati in una soluzione moderna, garantendo un'esperienza piacevole per gli utenti finali.

Examples

Comprensione dei file multimediali stand-alone

Il contenuto di esempio qui utilizzato è Tears of Steel, di Blender Foundation. Nello specifico, utilizzeremo il download intitolato "HD 720p (~ 365MB, mov, 2.0)". Questo è un singolo file che termina con l'estensione "mov" e giocherà praticamente in qualsiasi lettore multimediale moderno.

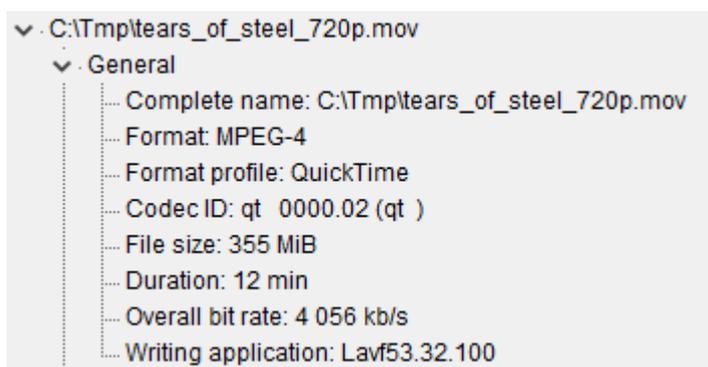
Si noti che la pagina di download offre i sottotitoli come download di file SRT separati. In questo esempio di contenuto, non ci sono sottotitoli consegnati insieme nello stesso file. Pertanto,

escludiamo l'analisi dei sottotitoli dall'ambito di questo esempio.

Un modo semplice per analizzare vari file multimediali consiste nell'utilizzare lo strumento / libreria [MediaInfo](#) . Mentre la funzionalità di analisi mostrata qui utilizza la GUI per semplicità, tutte le funzionalità sono disponibili anche tramite l'API MediaInfo.

Aperto questo file nella GUI di MediaInfo e passando alla visualizzazione ad albero, vedrai tre sezioni: Generale, Video e Audio. Il primo contiene le informazioni di base sul file, mentre i rimanenti due descrivono una traccia multimediale trovata in questo file. Esaminiamo le informazioni più rilevanti in ciascuna sezione dell'output.

Generale



I primi parametri di interesse sono il *formato* e il *profilo del formato* . Il primo indica che il **formato di imballaggio** proviene dalla suite di standard MPEG-4. MPEG-4 definisce il *formato di file multimediale di base ISO* e il *formato di imballaggio MP4* . Inoltre, Apple ha creato le proprie specifiche che derivano da queste, denominate in MediaInfo come profilo "QuickTime".

Nota: fare attenzione a non confondere MP4 e MPEG-4: il primo si riferisce a un formato di imballaggio specifico nella suite MPEG-4 di standard internazionali, che include anche i codec video e audio. Questo può portare a confusione, quindi evita di usare il termine MPEG-4 quando fai riferimento a qualcosa di diverso dal set completo di standard.

Tutti i formati di imballaggio basati sul formato di file multimediale di base ISO, definiti nella famiglia di standard MPEG-4, sono molto simili e spesso possono essere elaborati con gli stessi strumenti, le cui differenze sono in gran parte una questione di estensioni personalizzate dei fornitori che possono spesso essere sicure ignorate. Quindi, possiamo aspettarci che il video di esempio sia altamente compatibile con tutti i moderni lettori video.

video

```
Video
  ID: 1
  Format: AVC
  Format/Info: Advanced Video Codec
  Format profile: Main@L3.1
  Format settings, CABAC: No
  Format settings, ReFrames: 2 frames
  Format settings, GOP: M=4, N=18
  Codec ID: avc1
  Codec ID/Info: Advanced Video Coding
  Duration: 12 min
  Bit rate: 4 000 kb/s
  Width: 1 280 pixels
  Height: 534 pixels
  Display aspect ratio: 2.40:1
  Frame rate mode: Constant
  Frame rate: 24.000 FPS
  Color space: YUV
  Chroma subsampling: 4:2:0
  Bit depth: 8 bits
  Scan type: Progressive
  Bits/(Pixel*Frame): 0.244
  Stream size: 338 MiB (95%)
  Writing library: x264 core 122
  Encoding settings: cabac=0 / ref=2 / deblock=1:0:0 / ar
  Language: English
```

Il dettaglio più importante della traccia video è il codec utilizzato per trasformare i dati grezzi di colore in una forma compressa. Il nome del codec è fornito dal parametro *Format*.

AVC è anche noto come H.264 ed è il codec video che oggi è il più diffuso, supportato praticamente da tutti i moderni dispositivi e piattaforme software. Una traccia video codificata usando AVC sicuramente giocherà praticamente su qualsiasi giocatore.

I codec hanno spesso più *profili* che consentono di suddividere la funzionalità del codec in livelli, consentendo l'evoluzione della tecnologia in modo controllato. Il parametro del *profilo Format* indica che questo video utilizza il profilo principale. Questo profilo è relativamente raro, poiché quasi tutti i dispositivi moderni supportano il profilo alto, che offre una maggiore efficienza di compressione.

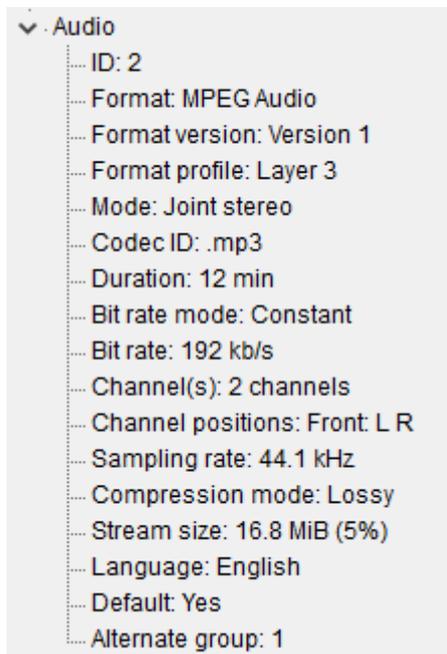
La qualità della traccia video è spesso di fondamentale importanza. Qui vediamo i fattori critici espressi dai parametri *Bit rate*, *Width* e *Height*. Gli ultimi due suggeriscono che si tratta di una traccia video a 720p, che è considerata una qualità HD di fascia bassa. L'immagine è in realtà più corta in verticale rispetto al frame 720p standard di 1280x720 pixel.

Il bit rate misura la quantità di dati che la forma compressa del flusso video occupa, in media, per ogni secondo di riproduzione. Questo è un parametro cruciale per l'ottimizzazione, in quanto la quantità di dati forniti è una delle principali fonti di costo nelle soluzioni video su larga scala.

I dati sopra riportati sulla qualità del video sono semplicemente fatti che otteniamo dall'analisi - qualsiasi giudizio sull'adeguatezza di questi parametri è un argomento che necessiterebbe di un'analisi molto più approfondita e viene affrontato da argomenti separati in questa categoria di

documentazione, come lo sono molti altri aspetti positivi di lavorando con tracce video.

Audio



Ancora una volta, la conoscenza del codec utilizzato per codificare i dati audio è di fondamentale importanza. Questo è espresso dai parametri di *formato* e *formato del profilo*. "MPEG Audio Layer 3" è più comunemente noto come MP3 ed è un formato audio universalmente supportato che ci si può aspettare che giochi ovunque.

Come per il video, i parametri di qualità audio sono i secondi punti dati più importanti, espressi principalmente dal parametro *Bit rate*.

Riepilogo delle analisi

Il contenuto è confezionato utilizzando un formato di packaging molto popolare, costruito sulla suite di standard MPEG-4. È codificato utilizzando i codec video e audio universalmente adottati. Da questo è chiaro che il video è pensato per essere facilmente accessibile a tutti gli spettatori - la compatibilità e la disponibilità sono state la chiave per i suoi autori.

L'uso di MP3 mostra l'età del contenuto dell'esempio, in quanto non è più considerato all'altezza dei concorrenti moderni - invece, AAC (Advanced Audio Coding) è il capofamiglia nel campo dei codec audio.

Lo stesso si può dire sull'utilizzo del profilo principale H.264. È molto raro che venga utilizzato qualsiasi profilo H.264 oltre a High, dato che quasi tutti i decoder lo supportano, consentendo a tutti di sfruttare l'efficienza migliorata abilitata dalle funzionalità di alto profilo.

I bitrate utilizzati sono leggermente più alti del previsto per l'ambiente odierno. Questo può essere spiegato dal desiderio degli autori di alta qualità o semplicemente dalle limitazioni degli encoder

che erano disponibili quando il contenuto è stato creato.

Di Più

Altri strumenti utili per l'analisi dei file multimediali sono [FFprobe](#), che fa parte del pacchetto software FFmpeg, e gli [strumenti di Bento4](#) per lavorare con i file MP4. Entrambi sono disponibili anche in forma di libreria. Sono in grado di eseguire analisi di livello più basso rispetto a MediaInfo, in situazioni in cui è necessario esaminare i singoli elementi che costituiscono i file multimediali.

Leggi Iniziare con il video online: <https://riptutorial.com/it/video/topic/5690/iniziare-con-il-video>

Capitolo 2: Capire una presentazione multimediale

Osservazioni

La riproduzione video è frontale e centrale in una vasta gamma di soluzioni moderne, con il software e gli standard in rapida evoluzione. Per capire cosa costituisce una presentazione multimediale, devi prima capire i molteplici aspetti coinvolti nel lavorare con il video:

- Le informazioni di colore grezzo acquisite dal mondo fisico sono comunemente *codificate* utilizzando un codec, un algoritmo il cui scopo è quello di rappresentare questi dati in una forma compressa, spesso sacrificando alcuni dettagli visivi in favore di una maggiore compressione.
- Per la riproduzione, viene eseguito l'algoritmo inverso - i dati vengono *decodificati* per diventare nuovamente informazioni di colore grezzo che possono essere fornite a un dispositivo di output (ad esempio un monitor).
- Tra codifica e decodifica, i dati compressi vengono *impacchettati* per l'archiviazione, il che può implicare la combinazione di tracce di tipi diversi in un singolo file o la segmentazione del contenuto in un numero elevato di segmenti piccoli.
- Il video viene *consegnato* al dispositivo dell'utente finale utilizzando una tecnologia di consegna, che può essere semplice come un download di file via HTTP o molto più complesso, che coinvolge feedback in diretta dall'infrastruttura di rete e adattamento automatico dei livelli di qualità.
- Il contenuto premium viene solitamente *crittografato* prima dell'imballaggio e può essere riprodotto solo su un lettore dotato di tecnologia DRM che garantisce la sicurezza della chiave di decodifica durante l'uso e protegge attivamente dall'acquisizione dell'output.

Mentre la porzione visiva è ovviamente dominante nella visibilità, l'audio e il testo svolgono anche un ruolo chiave nelle presentazioni multimediali, fornendo funzionalità multilingua che rendono i contenuti accessibili a un vasto pubblico. Nella maggior parte dei flussi di lavoro, le tracce audio e di testo vengono gestite in modo equivalente alle tracce video, vengono codificate, decodificate, pacchettizzate e consegnate lungo le stesse linee.

Tutti questi aspetti, e non solo, influenzano la composizione, la formattazione e l'uso di una presentazione multimediale. Devono essere compresi nel loro insieme al fine di utilizzare in modo efficace le tecnologie dei media.

Examples

Comprensione dei file multimediali stand-alone

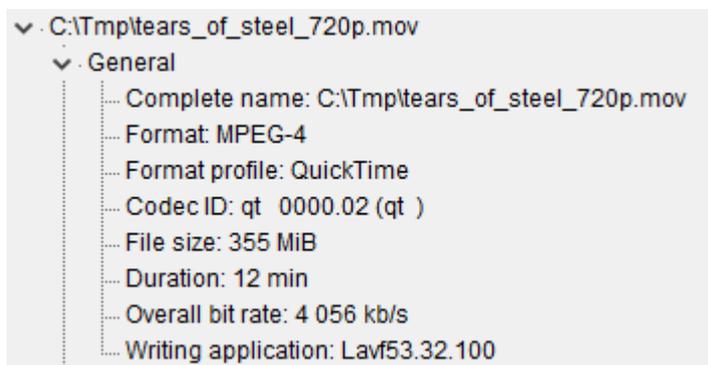
Il contenuto di esempio qui utilizzato è [Tears of Steel](#), di Blender Foundation. Nello specifico, utilizzeremo il download intitolato "HD 720p (~ 365MB, mov, 2.0)". Questo è un singolo file che termina con l'estensione "mov" e giocherà praticamente in qualsiasi lettore multimediale moderno.

Si noti che la pagina di download offre i sottotitoli come download di file SRT separati. In questo esempio di contenuto, non ci sono sottotitoli consegnati insieme nello stesso file. Pertanto, escludiamo l'analisi dei sottotitoli dall'ambito di questo esempio.

Un modo semplice per analizzare vari file multimediali consiste nell'utilizzare lo strumento / libreria [MedialInfo](#) . Mentre la funzionalità di analisi mostrata qui utilizza la GUI per semplicità, tutte le funzionalità sono disponibili anche tramite l'API MedialInfo.

Aperto questo file nella GUI di MedialInfo e passando alla visualizzazione ad albero, vedrai tre sezioni: Generale, Video e Audio. Il primo contiene le informazioni di base sul file, mentre i rimanenti due descrivono una traccia multimediale trovata in questo file. Esaminiamo le informazioni più rilevanti in ciascuna sezione dell'output.

Generale



I primi parametri di interesse sono il *formato* e il *profilo del formato* . Il primo indica che il **formato di imballaggio** proviene dalla suite di standard MPEG-4. MPEG-4 definisce il *formato di file multimediale di base ISO* e il *formato di imballaggio MP4* . Inoltre, Apple ha creato le proprie specifiche che derivano da queste, denominate in MedialInfo come profilo "QuickTime".

Nota: fare attenzione a non confondere MP4 e MPEG-4: il primo si riferisce a un formato di imballaggio specifico nella suite MPEG-4 di standard internazionali, che include anche i codec video e audio. Questo può portare a confusione, quindi evita di usare il termine MPEG-4 quando fai riferimento a qualcosa di diverso dal set completo di standard.

Tutti i formati di imballaggio basati sul formato di file multimediale di base ISO, definiti nella famiglia di standard MPEG-4, sono molto simili e spesso possono essere elaborati con gli stessi strumenti, le cui differenze sono in gran parte una questione di estensioni personalizzate dei fornitori che possono spesso essere sicure ignorato. Quindi, possiamo aspettarci che il video di esempio sia altamente compatibile con tutti i moderni lettori video.

video

```
Video
  ID: 1
  Format: AVC
  Format/Info: Advanced Video Codec
  Format profile: Main@L3.1
  Format settings, CABAC: No
  Format settings, ReFrames: 2 frames
  Format settings, GOP: M=4, N=18
  Codec ID: avc1
  Codec ID/Info: Advanced Video Coding
  Duration: 12 min
  Bit rate: 4 000 kb/s
  Width: 1 280 pixels
  Height: 534 pixels
  Display aspect ratio: 2.40:1
  Frame rate mode: Constant
  Frame rate: 24.000 FPS
  Color space: YUV
  Chroma subsampling: 4:2:0
  Bit depth: 8 bits
  Scan type: Progressive
  Bits/(Pixel*Frame): 0.244
  Stream size: 338 MiB (95%)
  Writing library: x264 core 122
  Encoding settings: cabac=0 / ref=2 / deblock=1:0:0 / ar
  Language: English
```

Il dettaglio più importante della traccia video è il codec utilizzato per trasformare i dati grezzi di colore in una forma compressa. Il nome del codec è fornito dal parametro *Format*.

AVC è anche noto come H.264 ed è il codec video che oggi è il più diffuso, supportato praticamente da tutti i moderni dispositivi e piattaforme software. Una traccia video codificata usando AVC sicuramente giocherà praticamente su qualsiasi giocatore.

I codec hanno spesso più *profili* che consentono di suddividere la funzionalità del codec in livelli, consentendo l'evoluzione della tecnologia in modo controllato. Il parametro del *profilo Format* indica che questo video utilizza il profilo principale. Questo profilo è relativamente raro, poiché quasi tutti i dispositivi moderni supportano il profilo alto, che offre una maggiore efficienza di compressione.

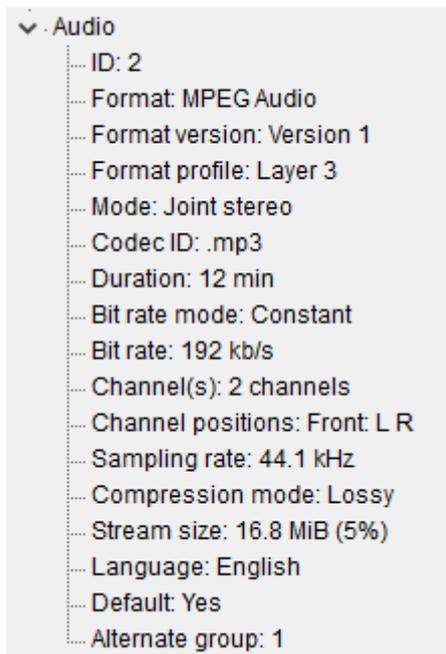
La qualità della traccia video è spesso di fondamentale importanza. Qui vediamo i fattori critici espressi dai parametri *Bit rate*, *Width* e *Height*. Gli ultimi due suggeriscono che si tratta di una traccia video a 720p, che è considerata una qualità HD di fascia bassa. L'immagine è in realtà più corta in verticale rispetto al frame 720p standard di 1280x720 pixel.

Il bit rate misura la quantità di dati che la forma compressa del flusso video occupa, in media, per ogni secondo di riproduzione. Questo è un parametro cruciale per l'ottimizzazione, in quanto la quantità di dati forniti è una delle principali fonti di costo nelle soluzioni video su larga scala.

I dati sopra riportati sulla qualità del video sono semplicemente fatti che otteniamo dall'analisi - qualsiasi giudizio sull'adeguatezza di questi parametri è un argomento che necessiterebbe di un'analisi molto più approfondita e viene affrontato da argomenti separati in questa categoria di

documentazione, come lo sono molti altri aspetti positivi di lavorando con tracce video.

Audio



Ancora una volta, la conoscenza del codec utilizzato per codificare i dati audio è di fondamentale importanza. Questo è espresso dai parametri di *formato* e *formato del profilo*. "MPEG Audio Layer 3" è più comunemente noto come MP3 ed è un formato audio universalmente supportato che ci si può aspettare che giochi ovunque.

Come per il video, i parametri di qualità audio sono i secondi punti dati più importanti, espressi principalmente dal parametro *Bit rate*.

Riepilogo delle analisi

Il contenuto è confezionato utilizzando un formato di packaging molto popolare, costruito sulla suite di standard MPEG-4. È codificato utilizzando i codec video e audio universalmente adottati. Da questo è chiaro che il video è pensato per essere facilmente accessibile a tutti gli spettatori - la compatibilità e la disponibilità sono state la chiave per i suoi autori.

L'uso di MP3 mostra l'età del contenuto dell'esempio, in quanto non è più considerato all'altezza dei concorrenti moderni - invece, AAC (Advanced Audio Coding) è il capofamiglia nel campo dei codec audio.

Lo stesso si può dire sull'utilizzo del profilo principale H.264. È molto raro che venga utilizzato qualsiasi profilo H.264 oltre a High, dato che quasi tutti i decoder lo supportano, consentendo a tutti di sfruttare l'efficienza migliorata abilitata dalle funzionalità di alto profilo.

I bitrate utilizzati sono leggermente più alti del previsto per l'ambiente odierno. Questo può essere spiegato dal desiderio degli autori di alta qualità o semplicemente dalle limitazioni degli encoder

che erano disponibili quando il contenuto è stato creato.

Di Più

Altri strumenti utili per l'analisi dei file multimediali sono [FFprobe](#), che fa parte del pacchetto software FFmpeg, e gli [strumenti di Bento4](#) per lavorare con i file MP4. Entrambi sono disponibili anche in forma di libreria. Sono in grado di eseguire analisi di livello più basso rispetto a MediaInfo, in situazioni in cui è necessario esaminare i singoli elementi che costituiscono i file multimediali.

Comprensione dei componenti delle tracce video

Questo esempio esplorerà come vedere il layout di una traccia video e come estrarre le singole immagini al suo interno.

Il contenuto di esempio qui utilizzato è [Tears of Steel](#), di Blender Foundation. Nello specifico, utilizzeremo il download intitolato "HD 720p (~ 365MB, mov, 2.0)". Questo è un singolo file che termina con l'estensione ".mov" e giocherà praticamente in qualsiasi lettore multimediale moderno.

Useremo gli strumenti mp4info e mp4dump della [suite Bento4](#) per il layout della traccia e l'analisi della struttura e [FFmpeg](#) per estrarre le singole immagini che compongono la traccia video.

Il film di esempio utilizza il formato di imballaggio "QuickTime" (MOV), basato sul formato di file multimediale di base ISO, uno standard internazionale che è alla base di tutti i formati di imballaggio nella famiglia di formati di file MP4. Ciò lo rende altamente compatibile con la maggior parte degli strumenti disponibili e consente un'analisi semplice.

Esaminiamo prima la struttura generale del file. Tutti i file multimediali basati su ISO Base Media File Format sono strutturati come una gerarchia di caselle: un mini-file system di sorta. Utilizzare l'utilità mp4dump per estrarre la struttura della scatola, eseguendo il seguente comando:

```
mp4dump tears_of_steel_720p.mov
```

L'output sarà simile al seguente:

```
[ftyp] size=8+12
  major_brand = qt
  minor_version = 200
  compatible_brand = qt
[wide] size=8+0
[mdat] size=8+371579623
[moov] size=8+598972
  [mvhd] size=12+96
    timescale = 1000
    duration = 734167
    duration(ms) = 734167
  [trak] size=8+244250
    [tkhd] size=12+80, flags=f
      enabled = 1
      id = 1
```

```
duration = 734167
width = 1280.000000
height = 534.000000
...
```

Questo rappresenta la struttura interna del file. Ad esempio, si vede qui una scatola *moov* che ha un header di 8 byte e 598972 byte di contenuti. Questa casella è un contenitore per varie caselle di metadati che descrivono il contenuto del file. Per ulteriori informazioni sul significato delle varie caselle e le loro proprietà, vedere [ISO / IEC 14496-12](#) .

Gli stessi campioni dei media stessi - le immagini compresse e le forme d'onda audio - sono memorizzati nella casella *mdat* , i cui contenuti sono opachi all'utilità mp4dump.

Per escludere dati irrilevanti e semplificare il flusso di lavoro dell'analisi - questo esempio si concentra sulla traccia video - ora rimuoviamo la traccia audio dal nostro filmato di esempio. Esegui il seguente comando:

```
ffmpeg -i tears_of_steel_720p.mov -an -vcodec copy video_track.mov
```

Nota che il passaggio precedente rimuoverà anche vari elementi di estensione personalizzati dal video di input, inserendo l'essenza del contenuto visivo in un nuovo file contenitore e scartando qualsiasi altra cosa. Se lo fai in uno scenario di produzione, assicurati di essere veramente libero di scartare tutti gli altri elementi nel file di input!

Le tracce video codificate sono una sequenza di immagini. Con il codec H.264 usato qui - e tutti gli altri codec moderni comunemente usati - le immagini possono essere di diversi tipi:

- I-immagini- sono immagini indipendenti, decodificabili utilizzando esclusivamente i dati contenuti all'interno dell'immagine.
- Immagini P - scattano un'altra immagine come linea di base e applicano una trasformazione a quell'immagine (es. "Sposta questi particolari pixel a destra di 5 pixel").
- Immagini B - simili ai fotogrammi P, ma bidirezionali - possono anche fare riferimento alle immagini del futuro e definire trasformazioni come "questi particolari pixel che saranno completamente visibili in 5 fotogrammi sono ora visibili al 10%".

L'esatta combinazione di tipi di immagini può essere scelta liberamente dal flusso di lavoro di codifica, creando molte opportunità di ottimizzazione, sebbene alcuni casi d'uso possano limitare la flessibilità disponibile, ad esempio richiedendo che un I-frame sia presente a intervalli di esattamente 2 secondi.

Esegui il seguente comando per vedere la struttura dell'immagine della traccia video:

```
mp4info --show-layout video_track.mov
```

Oltre a presentare una forma leggibile dagli utenti dei metadati di file generali, verrà visualizzata una stampa dettagliata del layout dell'immagine della traccia video.

```
...
00000959 [V] (1) size= 7615, offset=15483377, dts=491008 (39958 ms)
```

```

00000960 [V] (1) * size=104133, offset=15490992, dts=491520 (40000 ms)
00000961 [V] (1) size= 16168, offset=15595125, dts=492032 (40042 ms)
00000962 [V] (1) size=  4029, offset=15611293, dts=492544 (40083 ms)
00000963 [V] (1) size= 24615, offset=15615322, dts=493056 (40125 ms)
00000964 [V] (1) size=  4674, offset=15639937, dts=493568 (40167 ms)
00000965 [V] (1) size= 18451, offset=15644611, dts=494080 (40208 ms)
00000966 [V] (1) size= 95800, offset=15663062, dts=494592 (40250 ms)
00000967 [V] (1) size= 30271, offset=15758862, dts=495104 (40292 ms)
00000968 [V] (1) size= 10997, offset=15789133, dts=495616 (40333 ms)
00000969 [V] (1) size= 28458, offset=15800130, dts=496128 (40375 ms)
00000970 [V] (1) size=  9593, offset=15828588, dts=496640 (40417 ms)
00000971 [V] (1) size= 24548, offset=15838181, dts=497152 (40458 ms)
00000972 [V] (1) size=  6853, offset=15862729, dts=497664 (40500 ms)
00000973 [V] (1) size= 27698, offset=15869582, dts=498176 (40542 ms)
00000974 [V] (1) size=  7565, offset=15897280, dts=498688 (40583 ms)
00000975 [V] (1) size= 24682, offset=15904845, dts=499200 (40625 ms)
00000976 [V] (1) size=  5535, offset=15929527, dts=499712 (40667 ms)
00000977 [V] (1) size= 38360, offset=15935062, dts=500224 (40708 ms)
00000978 [V] (1) * size= 82466, offset=15973422, dts=500736 (40750 ms)
00000979 [V] (1) size= 13388, offset=16055888, dts=501248 (40792 ms)
00000980 [V] (1) size=  2315, offset=16069276, dts=501760 (40833 ms)
00000981 [V] (1) size= 21983, offset=16071591, dts=502272 (40875 ms)
00000982 [V] (1) size=  3384, offset=16093574, dts=502784 (40917 ms)
00000983 [V] (1) size= 22225, offset=16096958, dts=503296 (40958 ms)
...

```

Ogni riga in questa stampa è un'immagine contenuta nella traccia video. Quelli contrassegnati con un asterisco come (1) * sono immagini I. Puoi vedere come sono di dimensioni maggiori, con gli altri che consentono una maggiore compressione facendo riferimento alle immagini esistenti e descrivendo solo le differenze.

L'elenco contiene anche l'offset dei dati dell'immagine nel file video e il timestamp di decodifica dell'immagine, consentendo ulteriori correlazioni e analisi. Si noti che l'ordine di decodifica / tempistica delle immagini non è necessariamente lo stesso dell'ordine di presentazione / tempistica! Se le immagini B esistono nel video, possono essere decodificate solo *dopo* ogni immagine a cui fanno riferimento, anche se sono presentate *prima* delle immagini di riferimento!

Dopo aver acquisito alcune informazioni sulla struttura della traccia video, eseguire il seguente comando per estrarre le 30 immagini iniziando dal segno 40 secondi come file PNG:

```
ffmpeg -i video_track.mov -ss 00:00:40 -vframes 30 picture%04d.png
```

Le immagini estratte saranno completamente decodificate, come apparirebbero in un lettore video - non è possibile (senza strumenti estremamente specializzati) ottenere una rappresentazione visiva dei dati grezzi nei fotogrammi P o B.

Osserva come la settima immagine generata è un cambio di scena completo nel video. Si può facilmente correlare questo con l'output mp4info sopra - la settima immagine a partire dal segno 40 secondi (numero 00000966) è di dimensioni molto più grandi di quelle vicine. Le modifiche alle scene sono difficili da codificare, poiché aggiornano l'intera immagine e contengono molti nuovi dati. Se all'encoder non viene dato abbastanza trattamento per ottimizzare i cambiamenti di scena (ovvero non è consentito generare un'immagine di grandi dimensioni), l'output visivo sarà di bassa

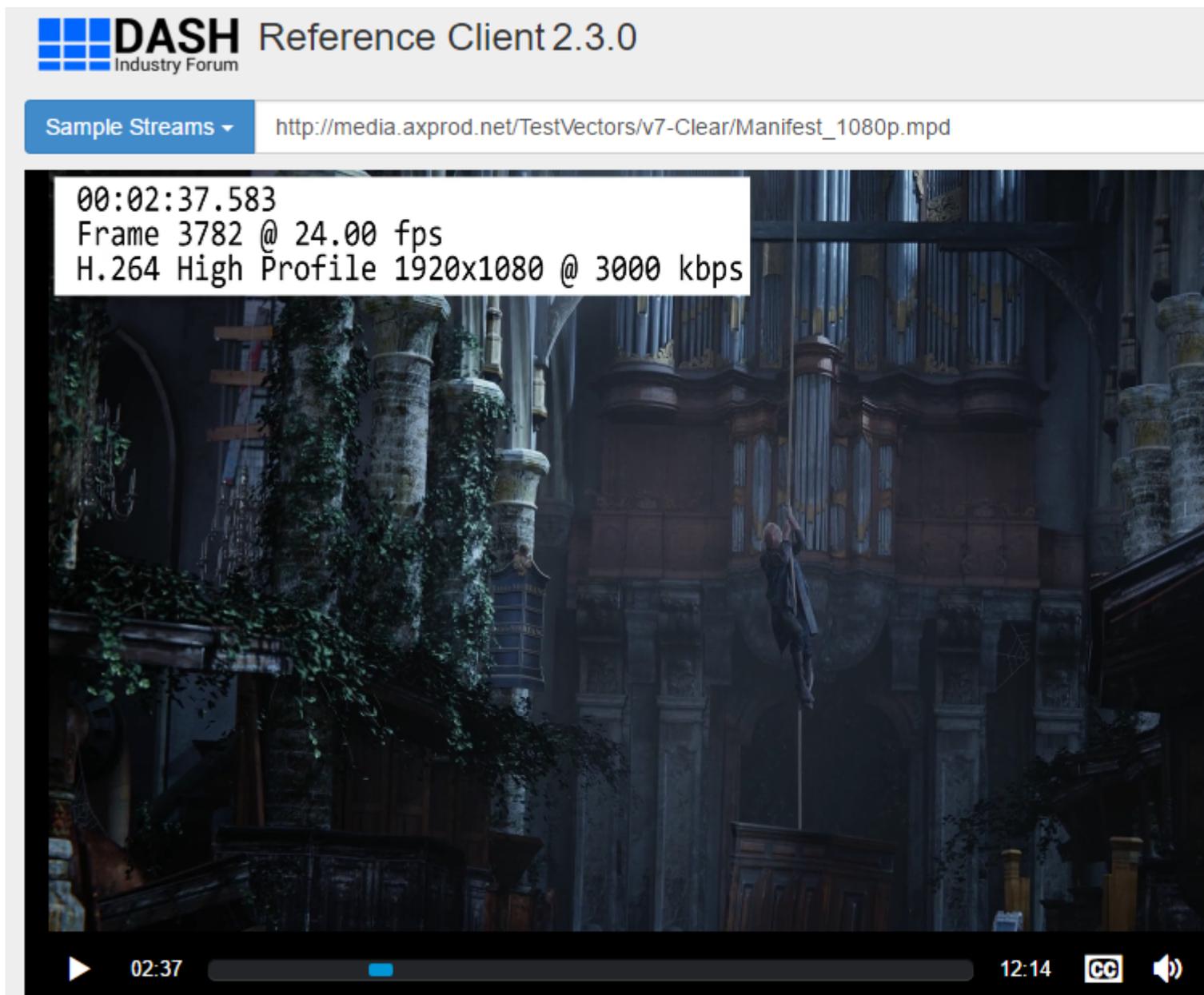
qualità o "blocky" fino alla successiva I-picture. Esaminando l'allocazione della larghezza di banda (byte) a varie immagini, è possibile ottenere informazioni su tali artefatti visivi che potrebbero apparire improvvisamente in un video.

Comprensione delle presentazioni di streaming adattive DASH

DASH è la tecnologia di streaming adattativo più diffusa in soluzioni moderne, utilizzata per fornire video in un'ampia varietà di scenari. Il modo migliore per comprendere le presentazioni DASH è osservare l'attività di rete che si svolge durante la riproduzione.

Questo esempio utilizza [Fiddler](#) per acquisire e analizzare il traffico di rete del browser, sebbene sia sufficiente qualsiasi strumento simile. Useremo il [player](#) open source [dash.js](#) per la riproduzione di video.

Per i nostri contenuti demo, utilizzeremo i [vettori di test DASH Axinom](#), in particolare la variante 1080p a singolo punto del vettore di test "Clear".



The screenshot displays the DASH Reference Client 2.3.0 interface. At the top left is the DASH Industry Forum logo. The main header reads "DASH Reference Client 2.3.0". Below this is a "Sample Streams" dropdown menu and a URL: `http://media.axprod.net/TestVectors/v7-Clear/Manifest_1080p.mpd`. The video player shows a scene from a movie with a man climbing a wall. A white text box in the upper left of the video frame displays the following metadata: `00:02:37.583`, `Frame 3782 @ 24.00 fps`, and `H.264 High Profile 1920x1080 @ 3000 kbps`. The video player controls at the bottom show a play button, a progress bar at 02:37, a total duration of 12:14, a Creative Commons license icon, and a speaker icon.

Con l'acquisizione della rete in esecuzione, apri il [dash.js nightly build sample player](#) in qualsiasi browser moderno e inserisci l'URL http://media.axprod.net/TestVectors/v6-Clear/Manifest_1080p.mpd nella casella di testo. Premere Carica per avviare la riproduzione. Osserverai i seguenti file scaricati:

```
http://media.axprod.net/TestVectors/v7-Clear/Manifest_1080p.mpd
http://media.axprod.net/TestVectors/v7-Clear/2/init.mp4
http://media.axprod.net/TestVectors/v7-Clear/15/init.mp4
http://media.axprod.net/TestVectors/v7-Clear/18/init.mp4
http://media.axprod.net/TestVectors/v7-Clear/18/0001.m4s
http://media.axprod.net/TestVectors/v7-Clear/2/0001.m4s
http://media.axprod.net/TestVectors/v7-Clear/15/0001.m4s
http://media.axprod.net/TestVectors/v7-Clear/15/0002.m4s
http://media.axprod.net/TestVectors/v7-Clear/5/init.mp4
http://media.axprod.net/TestVectors/v7-Clear/5/0002.m4s
http://media.axprod.net/TestVectors/v7-Clear/18/0002.m4s
http://media.axprod.net/TestVectors/v7-Clear/5/0003.m4s
...
```

Il primo file è il manifest di presentazione - un documento XML il cui formato è definito in [ISO / IEC 23009-1](#) . Questo descrive la presentazione DASH ad una profondità sufficiente a permettere al giocatore di capire come riprodurlo.

Se guardi all'interno del manifest, vedrai vari elementi di `AdaptationSet` , ognuno dei quali descrive un singolo adattamento del contenuto. Ad esempio, c'è un set di adattamento per la traccia video, tre set di adattamento per tre lingue audio e cinque set di adattamento per cinque lingue per i sottotitoli.

Gli insiemi di adattamento interni sono elementi di `Representation` . Per il set di adattamento video, ce ne sono diversi: ogni rappresentazione contiene lo stesso contenuto visivo codificato utilizzando un diverso livello di qualità. Ogni set di adattamento di audio e testo ha solo una rappresentazione.

```

AdaptationSet [ segmentAlignment=true group=1 maxWidth=1920 maxHeight=1080 maxFrameRate=24 par=16:9 lang=en ]
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" xmlns="urn:mpeg:dash:schema:mpd:2011" />
  <SegmentTemplate timescale="1200000" media="$RepresentationID$/$Number%04d$.m4s" startNumber="1" duration="1" />
  <Representation id="8" mimeType="video/mp4" codecs="hev1.2.4.L63.90" width="512" height="288" frameRate="24" />
  <Representation id="9" mimeType="video/mp4" codecs="hev1.2.4.L63.90" width="640" height="360" frameRate="24" />
  <Representation id="10" mimeType="video/mp4" codecs="hev1.2.4.L90.90" width="852" height="480" frameRate="24" />
  <Representation id="11" mimeType="video/mp4" codecs="hev1.2.4.L93.90" width="1280" height="720" frameRate="24" />
  <Representation id="12" mimeType="video/mp4" codecs="hev1.2.4.L120.90" width="1920" height="1080" frameRate="24" />
AdaptationSet [ segmentAlignment=true group=2 lang=en ]
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" xmlns="urn:mpeg:dash:schema:mpd:2011" />
  <SegmentTemplate timescale="24000" media="$RepresentationID$/$Number%04d$.m4s" startNumber="1" duration="1" />
  <Representation id="15" mimeType="audio/mp4" codecs="mp4a.40.29 audioSamplingRate=48000 startWithSAP=1" />
    <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="1" />
AdaptationSet [ segmentAlignment=true group=2 lang=en-AU ]
  <SegmentTemplate timescale="24000" media="$RepresentationID$/$Number%04d$.m4s" startNumber="1" duration="1" />
  <Representation id="16" mimeType="audio/mp4" codecs="mp4a.40.29 audioSamplingRate=48000 startWithSAP=1" />
    <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="1" />
AdaptationSet [ segmentAlignment=true group=2 lang=et-ET ]
  <SegmentTemplate timescale="24000" media="$RepresentationID$/$Number%04d$.m4s" startNumber="1" duration="1" />
  <Representation id="17" mimeType="audio/mp4" codecs="mp4a.40.29 audioSamplingRate=48000 startWithSAP=1" />
    <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="1" />
AdaptationSet [ segmentAlignment=true group=3 lang=en ]
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" xmlns="urn:mpeg:dash:schema:mpd:2011" />
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="subtitle" xmlns="urn:mpeg:dash:schema:mpd:2011" />
  <SegmentTemplate timescale="1000" media="$RepresentationID$/$Number%04d$.m4s" startNumber="1" duration="1" />
  <Representation id="18" mimeType="application/mp4" codecs="wvtt" startWithSAP="1" bandwidth="428" xmlns="urn:mpeg:dash:schema:mpd:2011" />

```

Per eseguire la riproduzione, un giocatore dovrà decidere quali set di adattamento presentare allo spettatore. Può prendere questa decisione in base a qualsiasi logica aziendale personalizzata o incorporata che desideri (ad esempio, l'ordine di preferenza della lingua). I set di adattamento che l'autore del contenuto considera primario hanno un elemento `Role` nel manifest che dichiara il ruolo "principale".

Inoltre, il giocatore dovrà decidere quale rappresentazione presentare allo spettatore (se un set di adattamento offre rappresentazioni multiple). La maggior parte dei giocatori inizia in modo conservativo e applica un algoritmo euristico che tenterà di presentare il livello di qualità massimo che la connessione di rete del visualizzatore può sostenere.

Il giocatore è libero di cambiare la serie attiva di rappresentazioni e / o set di adattamento in qualsiasi momento, in risposta all'azione dell'utente (selezionando una diversa linguetta) o alla logica automatica (l'euristica dell'ampiezza di banda determina un cambiamento del livello di qualità).

L'elemento `SegmentTemplate` definisce la struttura dell'URL che il giocatore può utilizzare per accedere alle diverse rappresentazioni. Un fattore chiave delle presentazioni DASH è che il contenuto viene suddiviso in piccoli segmenti di pochi secondi ciascuno (4 secondi nel caso del nostro filmato di esempio), che vengono scaricati in modo indipendente. Ogni rappresentazione ha anche un segmento di inizializzazione, denominato "init.mp4" per questo filmato di esempio, che contiene la configurazione del decodificatore specifica della rappresentazione e deve quindi essere caricato prima che qualsiasi altro segmento da quella rappresentazione possa essere elaborato.

Il comportamento qui descritto è accurato per il profilo DASH Live, che è la variante più comunemente usata di DASH. Esistono anche altri profili con un comportamento

leggermente diverso, non trattati qui. Presta attenzione all'attributo "profilo" sull'elemento radice manifest DASH per assicurarti che questa descrizione si applichi ai tuoi video!

Mentre esamini l'elenco di URL ottenuti dall'acquisizione del traffico di rete e li confronti con le informazioni fornite dal manifest, concluderai che il giocatore ha eseguito le seguenti azioni dopo aver scaricato il manifest:

1. Scarica i segmenti di inizializzazione per le rappresentazioni 2 (video a 360p), 15 (audio in inglese) e 18 (sottotitoli in inglese).
2. Scarica i primi segmenti delle tre rappresentazioni precedenti (0001.m4s).
3. Scarica il secondo segmento della rappresentazione audio.
4. Dal secondo segmento di video in poi, passa al flusso video 1080p! Ciò viene indicato scaricando il segmento di inizializzazione e il secondo segmento di rappresentazione 5 (video 1080p).
5. Continua a scaricare più segmenti delle rappresentazioni attive.

Osservando l'attività di rete, diventa facile osservare le decisioni prese da un riproduttore di streaming adattativo DASH. Un simile lettore è semplicemente un meccanismo che scarica segmenti di varie tracce e li fornisce in modo consecutivo a un motore di riproduzione multimediale, cambiando le tracce come appropriato.

I [vettori di test di Axinom DASH](#) contengono anche file di archivio che consentono di scaricare l'intera presentazione per l'analisi a livello di file system. Scoprirai che i file sul disco sono esattamente come sono a livello di rete. Ciò significa che le presentazioni DASH possono essere servite da server HTTP arbitrari, senza la necessità di alcuna logica personalizzata lato server.

Un aspetto del DASH del profilo live che complica l'analisi è che i campioni multimediali sono distribuiti su un numero elevato di segmenti. La maggior parte degli strumenti di analisi dei media non è in grado di elaborare singoli segmenti, operando solo su intere tracce. Spesso è possibile superare questa limitazione semplicemente concatenando i segmenti di una singola rappresentazione, iniziando dal segmento di inizializzazione. Ad esempio, su Windows è possibile utilizzare il seguente comando:

```
copy init.mp4 /b + 0001.mp4 /b + 0002.mp4 /b + 0003.mp4 /b track.mp4
```

Questo creerà un file track.mp4 che contiene i primi tre segmenti multimediali da una rappresentazione. Sebbene non sia identico nella struttura di un file MP4 autonomo, tale file può ancora essere analizzato dalla maggior parte degli strumenti (come mp4info e FFmpeg) senza una significativa perdita di funzionalità.

Leggi Capire una presentazione multimediale online:

<https://riptutorial.com/it/video/topic/5879/capire-una-presentazione-multimediale>

Capitolo 3: Proporzioni video

Osservazioni

I rapporti di aspetto sono spesso espressi come un rapporto larghezza: altezza che è spesso - ma non sempre - semplificato e talvolta anche come un semplice intero in virgola mobile.

Tutti i seguenti rapporti di aspetto sono lo stesso valore espresso in diversi modi:

- 1280: 720
- 16: 9
- 1,77777777777777777777777777777778

Examples

Visualizza proporzioni (DAR)



1280

720

Display Aspect Ratio (DAR) 16:9

Questo è uno screenshot di un video in riproduzione. Si vede un normale video in 16: 9 come ci si aspetterebbe di vedere in qualsiasi soluzione video moderna. Questo - il rapporto di aspetto che lo spettatore vede - è ciò che viene chiamato il *rapporto di aspetto dello schermo* o DAR.

Dai parametri illustrati, vediamo che $DAR = 1280: 720 = 16: 9 = 1.77777777777777777777777777777778$.

Formato immagine (PAR)

Internamente, tutti i video sono solo una serie di immagini. Diamo un'occhiata a una di queste immagini.



Sembra strano, vero? Infatti. Le immagini che compongono un video possono avere proporzioni diverse dal DAR, il più delle volte per motivi algoritmici (ad esempio, solo le dimensioni che sono un multiplo di 16 possono essere compresse dall'algoritmo scelto). Questo è chiamato *aspect ratio dell'immagine* o PAR.

In questo esempio, abbiamo le dimensioni dell'immagine esattamente invertite tra la forma visualizzata (16: 9) e l'immagine reale, quindi il PAR è 9:16. Normalmente, le differenze sono più piccole ma questo esempio è esagerato per chiarezza.

Dai parametri illustrati, vediamo che $PAR = 720: 1280 = 9:16 = 0.5625$

Rapporto di aspetto del campione (SAR)

Come indica l'esempio delle *proporzioni dell'immagine*, i video sono serie di immagini che non hanno necessariamente le stesse proporzioni del risultato finale da mostrare all'utente.

Quindi, come si ottiene da quelle immagini allungate all'output normalmente visualizzato? Hai bisogno di un fattore di allungamento! Questo fattore di allungamento viene applicato all'immagine per portarla alle proporzioni corrette per la visualizzazione. Questa è la *proporzione del campione* o SAR.



1280

Sample Aspect Ratio (SAR) 81:256



1280

720

Picture Aspect Ratio (PAR) 9:16

Il fattore di allungamento viene spesso espresso come un rapporto di due numeri interi. Lo si calcola come $SAR = PAR / DAR$.

Dai parametri illustrati, vediamo che $SAR = 9:16 / 16:9 = (9/16) / (16/9) = 81/256 = 3.1604938271604938271604938271605$

Proporzioni pixel

Questo è un altro nome per le *proporzioni del campione* e dovrebbe essere evitato, poiché l'acronimo naturale (PAR) è in conflitto con le *proporzioni dell'immagine*.

Leggi Proporzioni video online: <https://riptutorial.com/it/video/topic/5713/proporzioni-video>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con il video	Community , Sander
2	Capire una presentazione multimediale	Sander
3	Proporzioni video	Sander