

Guida a

 **Filmatch**

Frank Sinapsi

Guida a FFmatch

Frank Sinapsi

Indice

Introduzione

Esempio (15 ottobre 2023)

Approfondimenti (31 ottobre 2023)

Scorciatoie da tastiera (15 ottobre 2023)

Forzare collegamenti (15 ottobre 2023)

Spostamenti di sequenze (15 ottobre 2023)

Preferenze

Sequenze

Controlli

SIFT

Filtri

Misc

Problemi

Cambi “hard” di framerate

Difficoltà a stabilire i primi collegamenti

“Proportionality check failed”

Timestamps troppo irregolari (3 novembre 2020)

Compilazione

Introduzione

FFmatch è un programma open source che cerca di mettere in corrispondenza due versioni alternative dello stesso film, rilevando eventuali differenze interne.

Può essere usato per adattare a un video stream un audio stream proveniente da altra fonte.

Esempio (15 ottobre 2023)

In questa sezione è descritto l'utilizzo del programma in un caso reale che ho scelto appositamente per scopi didattici.

Ho scelto un film americano che è stato trasmesso da una emittente televisiva italiana (doppiato in italiano). A quanto ne so, di tale film non esiste il dvd in italiano, ma ho potuto recuperarne una registrazione televisiva in AVI/Xvid (d'ora in avanti la indicherò con il termine *tvrip*) e un *dvdrip* in lingua originale (d'ora in avanti lo indicherò con *dvdrip*), anch'esso in AVI/Xvid.

Il *tvrip* ha una qualità video abbastanza scadente, mentre la qualità del *dvdrip* è nettamente superiore. Si tratta quindi di un caso tipico in cui si vorrebbe “mettere l'audio italiano nel *dvdrip* in inglese”.

Inoltre il *tvrip* ha delle interruzioni pubblicitarie (che sono state tolte da chi lo ha codificato), e il *dvdrip* ha un framerate di 23.976 fps.

Il primo problema che si presenta è che il *tvrip* ha delle grosse bande nere sopra e sotto. Ecco un fotogramma:



Le bande nere possono avere effetti negativi sul riconoscimento, e al momento FFmpeg non ha un filtro di crop, quindi conviene preparare un nuovo filmato senza quelle grosse bande.

Per fare questo io procedo in questo modo, ma naturalmente ognuno può usare altri metodi equivalenti che gli sono più congeniali. Batto questa riga di comando:

```
mplayer "Un ragazzo come gli altri.avi" -vf cropdetect
```

Il filtro `cropdetect` di `mplayer` ha l'effetto di mostrare i valori di crop necessari per togliere le bande nere, e in questo caso ottengo le seguenti informazioni:

```
[CROP] Crop area: X: 0..479 Y: 92..482 (-vf crop=480:384:0:96).0 0
```

In genere per ottenere un crop più preciso è consigliabile controllare cosa viene croppato copiando i valori forniti da `mplayer` in un comando come questo:

```
mplayer "Un ragazzo come gli altri.avi" -vf rectangle=480:384:0:96
```

Ma in questo caso non c'è bisogno di andare tanto per il sottile, in quanto il filmato servirà esclusivamente per essere analizzato da `FFmatch`.

Quindi passo a generare la copia croppata, con questo comando:

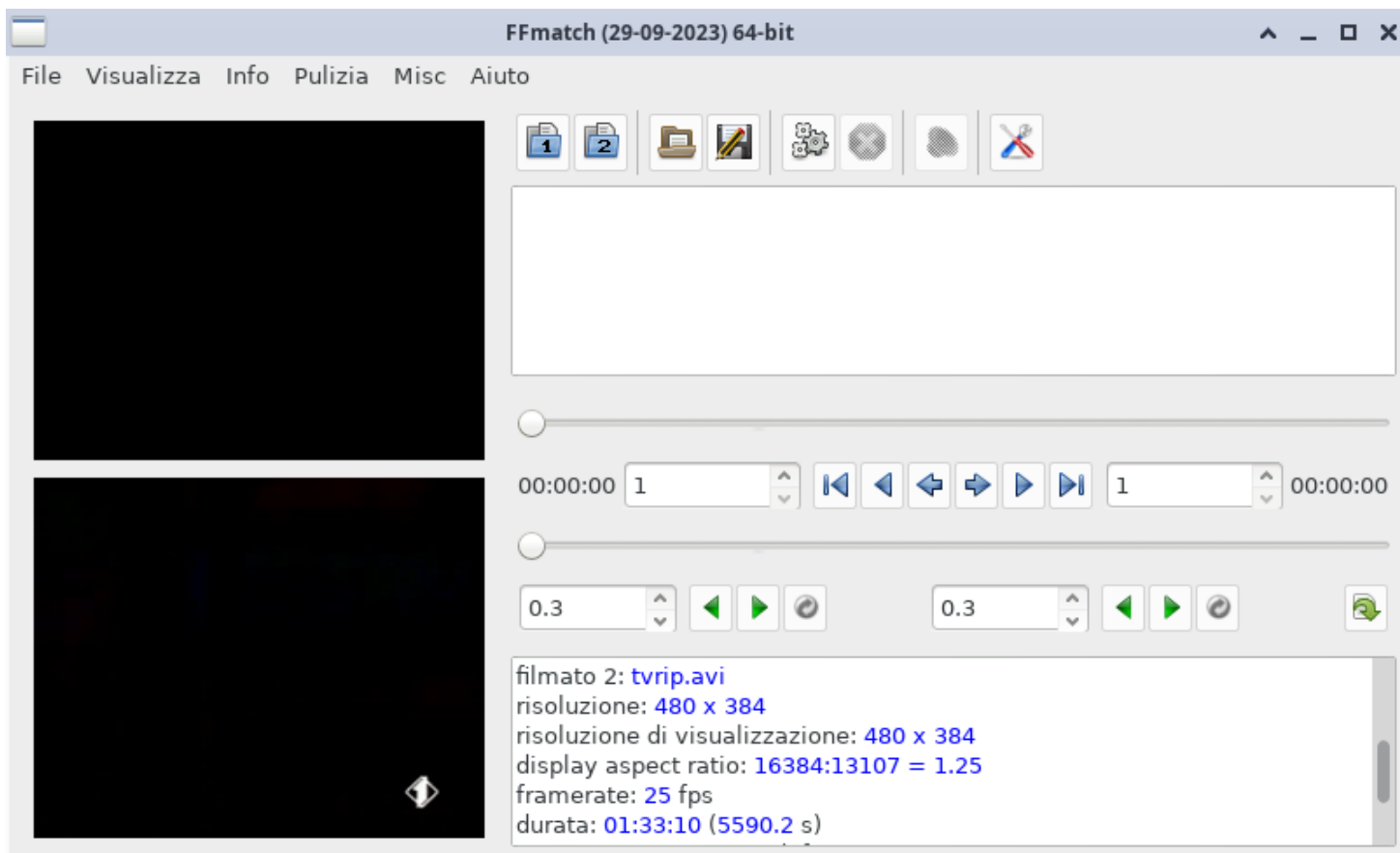
```
mencoder "Un ragazzo come gli altri.avi" -vf crop=480:384:0:96,harddup -ovc xvid -xvidencopts fixed_quant=2 -nosound -mc 0 -o tvrip.avi
```

Il comando precedente mi ha creato il file `tvrip.avi`, che è una versione AVI/Xvid croppata e senza audio del filmato di partenza.

Se uno dei filmati di partenza fosse troppo pesante da gestire, si può considerare se valga la pena procedere in modo analogo, creando una copia temporanea “usa e getta” in AVI/Xvid (eventualmente croppando e scalando a dimensioni inferiori, se ciò dovesse essere opportuno). Tale copia servirà solo per essere analizzata da `FFmatch`, dopodiché potrà essere cancellata.

Ciò che conta è che i frame dei filmati dati in pasto a `FFmatch` siano nelle stesse posizioni dei filmati originali.

Dopo questo lavoro preparatorio, si può lanciare `FFmatch` e selezionare i due filmati usando i primi due pulsanti a sinistra, nella barra dei pulsanti in alto.



La selezione di entrambi i filmati ha come effetto l'abilitazione del quinto pulsante (quello con le rotelline), che serve per far partire l'analisi; non resta che cliccarci sopra e aspettare che il programma si fermi.

Nel mio caso, quando il processo si arresta, nel testo di sopra ottengo queste righe:

```
(1) Just One Of The Guys 1985 DvDrip[Eng]-greenbud1969.avi
(2) tvrip.avi

free (1): 5695/144327
free (2): 1679/139753

controlli: 1644 (passati: 1614)
rapporto orizzontale: 0.99
rapporto verticale: 1.031
internal diff: 11
free segm: 9

[000548-025546] ---> [000001-024999] {024999} (diff=-547)
[025547-025589] -/-> ??? {43-38}
[025590-027504] ---> [025038-026952] {001915} (diff=-552)
[027505-027834] -/-> ??? {330-327}
[027835-049744] ---> [027280-049189] {021910} (diff=-555)
[049748-054522] ---> [049190-053964] {004775} (diff=-558)
[054525-055658] ---> [053965-055098] {001134} (diff=-560)
[055662-072745] ---> [055099-072182] {017084} (diff=-563)
[072746-072757] -/-> ??? {12-6}
[072758-077570] ---> [072189-077001] {004813} (diff=-569)
[077571-077702] -/-> ??? {132-128}
[077703-078651] ---> [077130-078078] {000949} (diff=-573)
[078652-078675] -/-> ??? {24-37}
[078676-085230] ---> [078116-084670] {006555} (diff=-560)
```

```
[085231-085413] -/-> ??? {183-179}
[085414-109621] ---> [084850-109057] {024208} (diff=-564)
[109622-109687] -/-> ??? {66-41}
[109688-116406] ---> [109099-115817] {006719} (diff=-589)
[116410-127897] ---> [115818-127305] {011488} (diff=-592)
[127898-127921] -/-> ??? {24-24}
[127922-139446] ---> [127330-138854] {011525} (diff=-592)
[139447-144327] -/-> ??? {4881-899}
```

frame(s) del primo filmato che non sono presenti nel secondo:

```
[000001-000547]1---> [000000] {547}
[049745-049747]1---> [049189] {3}
[054523-054524]1---> [053964] {2}
[055659-055661]1---> [055098] {3}
[116407-116409]1---> [115817] {3}
```

Le prime due righe riportano i nomi dei filmati analizzati.

Nelle successive due righe sono riportati il numero di frame che sono rimasti liberi (non collegati) sul totale, per entrambi i filmati.

Segue il numero totale di controlli effettuati, e quanti di questi hanno dato esito positivo. Un controllo è un tentativo di mettere in corrispondenza un certo frame di un filmato con un certo frame dell'altro filmato. FFmatch non effettua controlli alla cieca, ma è guidato da uno schema di pattern matching applicato alle lunghezze delle scene. Quindi la maggior parte dei controlli effettuati ha un'alta probabilità di dare esito positivo. Per questo il numero di controlli falliti è di solito relativamente basso.

I rapporti orizzontale e verticale sono i fattori di espansione (se il rapporto è maggiore di 1) o di contrazione (se minore di 1) tra il primo e il secondo filmato. Tali rapporti non coincidono necessariamente con i rapporti tra larghezze e altezze dei due filmati, perché questi potrebbero avere subito un cropping differente.

Il campo *internal diff* indica il numero di differenze interne che sono state al momento rilevate. Il campo *free segm* indica il numero di righe rosse, cioè il numero di segmenti interni che al momento non sono stati ancora collegati.

L'elenco successivo è la *mappatura* tra i due filmati, cioè la sequenza degli intervalli nei quali il programma ha ripartito i filmati in base alle differenze interne rilevate (un *intervallo*, o *zona*, è una sequenza di frame consecutivi di un filmato).

Nel riquadro gli intervalli vengono mostrati in due colori diversi.

Gli intervalli in verde sono quelli che il programma ha collegato a intervalli dell'altro filmato; si noti che due intervalli collegati hanno lo stesso numero di frames, indicato tra parentesi graffe.

Gli intervalli in rosso sono intervalli del primo filmato che per qualche motivo il programma ha rinunciato a collegare. Sono quindi una parte dei frame riportati come liberi.

Possono esserci altri frame liberi, non riportati nelle righe rosse. Gli intervalli in rosso che vengono mostrati sono solo quelli in cui per entrambi i filmati c'è almeno un frame libero; le ampiezze di tali intervalli sono indicate tra parentesi graffe, nella forma $\{n_1-n_2\}$, dove n_1 e n_2 sono entrambi strettamente positivi.

Per rendere la mappatura più facile da leggere, gli intervalli liberi di un filmato che corrispondono a un intervallo vuoto dell'altro non vengono mostrati come righe rosse (dove una delle due lunghezze sarebbe stata 0), ma nelle sezioni successive del rapporto, come “frame di un filmato che non sono presenti nell'altro”.

Per ottenere buoni risultati, è indispensabile capire bene le notazioni usate nel rapporto. Una spiegazione più dettagliata viene data nella prossima sezione.

Un'altra cosa importante da capire è che la nuova traccia audio verrà costruita partendo dalla traccia audio in inglese, **sostituendo a questa tutte le zone verdi con le corrispondenti zone della traccia audio in italiano**.

Questo vuol dire, per fare un esempio, che se nel rapporto non c'è nemmeno una riga verde, la traccia ricostruita sarà una copia esatta della traccia in inglese.

Tornando all'esempio, possiamo notare che il programma ha mappato un alto numero di intervalli, ma ne ha anche lasciati alcuni non risolti (per l'esattezza, 8 interni e uno finale).

A questo punto si entra nella fase più impegnativa per l'utente: l'eliminazione di eventuali errori di mappatura e il completamento delle parti non risolte.

Gli strumenti per modificare la mappatura sono questi:



Tralasciando i due slide orizzontali e le due etichette che indicano le rispettive posizioni (nella figura 00:00:00), nella prima riga abbiamo, partendo da sinistra:

- uno spin button numerico (nella figura c'è il valore 1);
- sei frecce blu, tre rivolte verso sinistra e tre verso destra;
- un secondo spin button numerico.

Nella seconda riga abbiamo:

- un primo gruppo costituito da uno spin button numerico, due frecce verdi e un pulsante (che chiamo *primo pulsante di match*);
- un secondo gruppo identico al primo (con un *secondo pulsante di match*);
- un pulsante finale, che in seguito chiamerò *pulsante di collegamento*.

Le caratteristiche fondamentali sono le seguenti:

- tutti questi strumenti, con la sola eccezione del pulsante di collegamento, permettono soltanto di fare spostamenti sui filmati (e uno spostamento non modifica in alcun modo la mappatura corrente);
- il pulsante di collegamento è l'unico strumento che permette di modificare la mappatura, e soltanto dopo aver chiesto conferma all'utente;
- le frecce blu hanno la caratteristica di lasciare sempre inalterata la differenza tra le posizioni dei due filmati.

Gli spostamenti sono di due tipi: *sincroni* (o *vincolati*) e *svincolati*.

Uno spostamento è sincrono se lascia inalterata la differenza tra le posizioni. Quindi l'ultima caratteristica può essere riformulata così: "le frecce blu effettuano solo spostamenti sincroni".

Inoltre, le frecce blu sono gli unici strumenti che effettuano spostamenti sincroni.

Uno spostamento è svincolato se ha effetto sulla posizione di uno solo dei due filmati.

Ci sono tre tipi di frecce blu. Quelle al centro effettuano spostamenti di una singola posizione; poi ci sono quelle che spostano di 10 posizioni (l'entità degli spostamenti può essere modificata dalla finestra delle preferenze). Infine, le frecce blu laterali spostano sugli estremi dell'intervallo corrente.

I due spin della riga superiore permettono di fare spostamenti svincolati sui filmati, e mostrano sempre le posizioni correnti.

Le frecce verdi della riga inferiore permettono di saltare tra le scene dei filmati: cliccando sulla prima freccia verde rivolta a destra si salta al frame in cui inizia la prossima scena nel primo filmato; e così via...

Gli spin situati alla sinistra delle frecce verdi sono soglie che servono per regolare il rilevamento degli inizi di scena. Aumentando tali soglie si otterranno salti più ampi, perché verranno considerati "cambi di scena" solo quei frame che differiscono molto dal frame che li precede (in base al valore di soglia impostato); abbassando tali soglie fino al valore minimo (0), si otterranno salti sempre più piccoli fino a spostamenti di un singolo frame (con il valore minimo, cioè 0, ogni frame viene considerato un "cambio di scena").

Il primo pulsante di match può avere come effetto uno spostamento soltanto sul secondo filmato. Se il frame corrente del primo filmato è attualmente mappato (cioè, non libero), cliccando sul pulsante di match si otterrà uno spostamento del secondo filmato sul frame attualmente collegato a quello del primo filmato. In questo modo l'utente può controllare se il frame mostrato di sopra coincide con quello mostrato di sotto.

Ma la vera utilità del pulsante di match si ha quando lo si usa sui frame liberi; in tal caso, se possibile, il programma posizionerà il secondo filmato sui frame che ritiene più probabili, in base a considerazioni su allineamenti. Cliccando più volte sul pulsante di match è possibile che il programma si

posizioni (a rotazione) sui frame che più probabilmente corrispondono al corrente frame del primo.

Per il secondo pulsante di match vale quanto detto per il primo, invertendo “primo” e “secondo”.

Oltre agli spostamenti già discussi, esistono altri due modi più immediati per spostarsi nei filmati: il primo è quello di trascinare i rispettivi slide orizzontali (si tratta di spostamenti svincolati); il secondo è quello di **clickare sopra i numeri di colore blu** nel testo di sopra. Si noti che i numeri di colore blu sono le uniche porzioni di testo che consentono, cliccandoci sopra, di fare spostamenti sui filmati.

Il pulsante di collegamento, come ho già detto, è l’unico che permette di modificare la mappatura corrente.

Il suo funzionamento è il seguente: se si clicca sul pulsante di collegamento quando il frame corrente del primo filmato è libero, il programma lo interpreterà come una richiesta di creazione del collegamento tra tale frame e il frame corrente del secondo filmato. Se la creazione di tale collegamento è possibile, il programma chiederà conferma all’utente, e in caso di risposta affermativa creerà il collegamento.

Se, invece, i due frame correnti sono attualmente già collegati tra loro, il programma chiederà all’utente se desidera che il collegamento venga rimosso.

In quest’ultimo caso occorre fare molta attenzione, perché il collegamento che viene rimosso è in realtà un collegamento tra due intervalli che contengono i due frame correnti, e tipicamente tali intervalli comprenderanno altri frame oltre a quei due.

Maggiori dettagli sul pulsante di collegamento sono dati nella sezione [Forzare collegamenti](#).

La tecnica base per completare una mappatura consiste nel clickare sui numeri blu, uno dietro l’altro. Nel caso di righe verdi, si controlla se il collegamento è corretto, altrimenti lo si rimuove.

Nel caso di righe rosse occorre decidere, di volta in volta, come conviene “chiudere” la mappatura. In casi molto particolari potrebbe anche essere preferibile lasciare degli intervalli non risolti.

Giocando un po’ con il programma, l’utente imparerà alcuni “metodi” per svolgere velocemente questo compito. Nelle prossime sezioni ne verranno

discussi alcuni. Per dare una prima idea, una strategia consiste nello stabilire collegamenti in posizioni cruciali, e far partire nuovamente l'analisi. Se i collegamenti forzati manualmente vengono scelti in modo opportuno, gli algoritmi di analisi li sfrutteranno per portare avanti la creazione di nuovi collegamenti. Questo metodo può essere applicato più volte di seguito. Occorre però fare attenzione al fatto che, se si sono rimossi dei collegamenti sbagliati e si fa partire nuovamente l'analisi, è abbastanza probabile che il programma crei nuovamente gli stessi collegamenti sbagliati. Quindi è consigliabile forzare collegamenti corretti e "cruciali" subito dopo aver rimosso un collegamento sbagliato; con "cruciali" intendo collegamenti che impediscano al programma di ricreare il collegamento sbagliato, nel caso in cui venga fatta ripartire l'analisi.

Applicando all'esempio i metodi accennati sopra (come già detto, nelle prossime sezioni vengono dati maggiori dettagli), mi sono portato in questa situazione:

```
(1) Just One Of The Guys 1985 DvDrip[Eng]-greenbud1969.avi
(2) tvrip.avi
```

```
free (1): 4886/144327
free (2): 904/139753
```

```
controlli: 1644 (passati: 1614)
rapporto orizzontale: 0.99
rapporto verticale: 1.031
internal diff: 12
free segm: 1
```

```
[000548-025566] ---> [000001-025019] {025019} (diff=-547)
[025572-027573] ---> [025020-027021] {002002} (diff=-552)
[027576-027745] ---> [027022-027191] {000170} (diff=-554)
[027747-049649] ---> [027192-049094] {021903} (diff=-555)
[049653-054522] ---> [049095-053964] {004870} (diff=-558)
[054525-055658] ---> [053965-055098] {001134} (diff=-560)
[055662-072747] ---> [055099-072184] {017086} (diff=-563)
[072754-077636] ---> [072185-077067] {004883} (diff=-569)
[077641-078665] ---> [077068-078092] {001025} (diff=-573)
[078666-085252] ---> [078106-084692] {006587} (diff=-560)
[085257-109639] ---> [084693-109075] {024383} (diff=-564)
[109665-116406] ---> [109076-115817] {006742} (diff=-589)
[116410-139441] ---> [115818-138849] {023032} (diff=-592)
[139442-144327] -/-> ??? {4886-904}
```

```
frame(s) del primo filmato che non sono presenti nel secondo:
```

```
[000001-000547]1---> [000000] {547}
[025567-025571]1---> [025019] {5}
[027574-027575]1---> [027021] {2}
[027746-027746]1---> [027191] {1}
[049650-049652]1---> [049094] {3}
[054523-054524]1---> [053964] {2}
[055659-055661]1---> [055098] {3}
[072748-072753]1---> [072184] {6}
[077637-077640]1---> [077067] {4}
[085253-085256]1---> [084692] {4}
```

```
[109640-109664]1---> [109075] {25}
```

```
[116407-116409]1---> [115817] {3}
```

frame(s) del secondo filmato che non sono presenti nel primo:

```
[078093-078105]2---> [078665] {13}
```

Ho deciso di lasciare l'ultimo intervallo non risolto, perché pochi frame dopo l'estremo superiore dell'ultima riga verde iniziano i titoli di coda. Dato che questi sono troncati nel tvrip (si può notare che nel tvrip sono rimasti solo 904 frames contro i 4886 del dvdrip) e che da lì in poi non ci sono più dialoghi ma solo musica, lasciando l'ultima riga rossa si otterrà come risultato il passaggio dal doppiato italiano alla musica del dvdrip appena un po' prima dell'inizio dei titoli di coda.

A questo punto la mappatura può essere considerata completa. Non resta che generare la traccia audio.

Per prima cosa estraggo le tracce audio dai filmati di partenza con questi comandi:

```
ffmpeg -i "Un ragazzo come gli altri.avi" ita44100.wav
```

```
ffmpeg -i "Just One Of The Guys 1985 DvDrip[Eng]-greenbud1969.avi" eng48000.wav
```

I comandi creano i file `ita44100.wav` e `eng48000.wav`, che chiamo così per ricordarmi le relative frequenze di campionamento che in questo caso sono 44100 Hz (traccia italiana) e 48000 Hz (traccia inglese).

In alternativa, avrei potuto decomprimere le tracce in formato SOX non compresso, analogo a WAV non compresso ma con meno limitazioni (i file SOX possono essere più grandi di 4 GB e a volte può risultare utile quando si ha a che fare con film lunghi e tracce audio con molti canali).

Ciò che conta è che le tracce (WAV o SOX) da inserire in FFmatch **devono essere tra loro omogenee**, cioè devono essere codificate nello stesso formato, avere la stessa frequenza di campionamento, lo stesso numero di canali e la stessa dimensione, in bit, dei campioni.

In questo caso la frequenza di campionamento è diversa, quindi una delle due tracce va necessariamente ricampionata.

Inoltre devo tenere in considerazione che il dvdrip ha un frame rate di 23.976 fps e il tvrip di 25 fps.

La mia scelta è di creare un filmato con frame rate 25 fps, usando ovviamente il video del dvdrip.

Guardando la mappatura è evidente che la traccia audio in italiano sarà costituita per la maggior parte dall'audio del tvrip, che quindi è già pronto per essere usato (file `ita44100.wav`).

Resta quindi soltanto da preparare una traccia WAV non compressa a 44100 Hz del dvdrip, tenendo conto del fatto che il video verrà accelerato da 23.976 a 25 fps.

Va pertanto fatto un resampling, tenendo conto di due vincoli:

1. devo ottenere un WAV a 44100 Hz;
2. il video verrà accelerato di un fattore 25/23.976.

Dato che moltiplicando 48000 per $25/(24000/1001)$ si ottiene 50050, il resampling che mi serve può essere ottenuto con questo comando:

```
sox -r 50050 eng48000.wav -r 44100 eng44100.wav
```

Si noti che, sotto Linux, invece di usare ffmpeg e sox in tempi separati, avrei potuto combinarli in una pipeline come segue, evitando così di generare il file intermedio `eng48000.wav`:

```
ffmpeg -i "Just One Of The Guys 1985 DvDrip[Eng]-greenbud1969.avi" -f wav - | sox -r 50050 -t wav -  
-r 44100 eng44100.wav
```

Apro una parentesi: e se avessi scelto di ottenere un filmato a 23.976 fps, come avrei dovuto procedere? In tal caso avrei potuto lasciare invariata la traccia inglese e ricampionare la traccia italiana in questo modo:

```
sox -r 42293.706293 ita44100.wav -r 48000 ita48000.wav
```

Il valore 42293.706293 si ottiene moltiplicando la frequenza iniziale (44100) per 24000/1001 (cioè 23.976) e dividendo per 25. Parentesi chiusa.

Tornando alla nostra scelta iniziale, le tracce che abbiamo ottenuto (`ita44100.wav` e `eng44100.wav`) hanno tutte le caratteristiche tecniche per essere inserite in FFmatch.

C'è però un'altra operazione che possiamo fare, per renderle ancora più omogenee: normalizzarne i volumi, ad esempio con questi comandi:

```
wavegain -y ita44100.wav  
wavegain -y eng44100.wav
```

Una tale normalizzazione è tanto più consigliata quanto più estese sono le parti di audio inglese che non verranno sostituite dalla traccia italiana (ad esempio, se il dvdrip avesse scene reintegrate che mancano nel tvrip e che pertanto resteranno non doppiate, oppure se abbiamo lasciato la musica dei titoli di testa e/o di coda, come in questo caso). Se non facciamo la normalizzazione può succedere che una delle due tracce abbia un volume molto più alto dell'altra e tale differenza potrà risultare fastidiosa nei punti in cui si passa dall'una all'altra.

Adesso ho tutto ciò che serve per generare la traccia in italiano, quindi clicco sul pulsante con l'altoparlante e seleziono le due tracce WAV che ho preparato (eng44100.wav per il primo filmato e ita44100.wav per il secondo); inoltre seleziono l'opzione "video 1+audio 2", dato che voglio adattare l'audio del tvrip (filmato 2) al video del dvdrip (filmato 1):



Infine clicco sul pulsante salva la traccia come... e salvo la nuova traccia WAV con il nome da-muxare.wav.

Il file che ho ottenuto (da-muxare.wav) ha la stessa dimensione (in bytes) del file eng44100.wav. Non si tratta di un caso poiché, come già detto in precedenza, **la traccia audio generata da FFmatch è una copia dell'audio in inglese in cui sono state sostituite le zone verdi con le corrispondenti zone in italiano.**

L'ultima cosa che resta da fare è ricomprimere e muxare l'audio ottenuto al dvdrip, ricordando che quest'ultimo va accelerato da 23.976 a 25 fps.

Questa operazione può essere facilmente fatta usando uno tra i tanti programmi esistenti.

Ad esempio, un comando come questo comprime la traccia in formato AC3:

```
ffmpeg -i da-muxare.wav da-muxare.ac3
```

Dopodiché si può procedere a muxare la traccia `da-muxare.ac3` al `dvdrrip` con un qualsiasi programma di muxing, avendo cura di modificare il frame rate nominale da 23.976 a 25 fps (si noti che una modifica del frame rate nominale non comporta alcuna ricodifica dello stream video, ma solo una modifica dei timestamps dei fotogrammi).

Approfondimenti (31 ottobre 2023)

In questa sezione fornirò dettagli che ho tralasciato nella precedente sezione per non appesantirla troppo.

Consideriamo una qualsiasi riga verde della mappatura, ad esempio la seguente:

```
[054525-055658] ---> [053965-055098] {001134} (diff=-560)
```

Il significato dei primi due numeri di colore blu e dei primi due numeri di colore verde dovrebbe essere ovvio: si tratta degli estremi dell'intervallo del primo filmato [054525-055658] che sono stati collegati all'intervallo del secondo filmato compreso fra tali estremi [053965-055098].

Il numero tra parentesi graffe {001134} è l'ampiezza di tali intervalli (55658+1-54525).

Il numero tra parentesi tonde alla fine (diff=-560) è la differenza tra le rispettive posizioni all'interno di ciascun filmato (53965-54525).

E questo è tutto, per quanto riguarda le righe verdi.

Adesso consideriamo una riga rossa della mappatura, ad esempio la seguente (per chiarezza riporto anche la riga verde che la precede e quella che la segue):

```
[025590-027504] ---> [025038-026952] {001915} (diff=-552)  
[027505-027834] -/-> ??? {330-327}  
[027835-049744] ---> [027280-049189] {021910} (diff=-555)
```

Il significato dei primi due numeri di colore blu è lo stesso che hanno nelle righe verdi, ma la riga rossa è caratterizzata da una freccia spezzata (-/->) che sta a indicare che il segmento non è collegato e da una coppia di numeri tra parentesi graffe, separati da un trattino: {n-m}.

I numeri tra parentesi graffe indicano le ampiezze (numero di fotogrammi) dei segmenti scollegati del primo e del secondo filmato, rispettivamente.

In questo caso, i due numeri tra parentesi graffe sono diversi: {330-327}.

Quando i numeri sono diversi, come in questo caso, abbiamo a che fare con una *differenza interna*.

Si può dire che uno degli scopi principali di FFmatch è quello di rilevare (possibilmente tutte) le differenze interne tra i due filmati.

Le ho chiamate *differenze interne* per distinguerle dalle differenze riportate alla fine delle righe verdi: in quel caso, come già detto, si tratta della differenza tra le posizioni di una stessa sequenza nei due filmati; una differenza interna è invece una differenza tra quelle differenze.

Per chiarire le idee, osserviamo che in questa sequenza irrisolta il primo filmato ha 3 fotogrammi in più del secondo (330-327). Nella riga verde precedente abbiamo (diff=-552). Nella riga verde successiva abbiamo (diff=-555). La differenza è cambiata in conseguenza della differenza interna data dalla riga rossa {330-327}.

Ovviamente, se vogliamo che audio e video siano ovunque ben sincronizzati, dobbiamo gestire in modo opportuno ciascuna differenza interna.

Più è alto il numero di differenze interne rilevate, più è difficile arrivare a ottenere una mappatura completa e corretta.

Possiamo dire che il campo *Internal diff* del rapporto mostrato nel testo di sopra è il primo da considerare, alla fine di un riconoscimento automatico, per rendersi conto della difficoltà del mux.

Ci sono casi in cui è possibile ridurre facilmente le differenze interne di minima entità (cioè, quello del tipo {n-m} dove n e m differiscono in valore assoluto di 1).

Per prima cosa occupiamoci delle righe rosse del tipo {n-n}. Consideriamo ad esempio questa:

```
[116410-127897] ---> [115818-127305] {011488} (diff=-592)
[127898-127921] -/-> ??? {24-24}
[127922-139446] ---> [127330-138854] {011525} (diff=-592)
```

Qui abbiamo un segmento irrisolto lungo 24 fotogrammi (circa 1 secondo di filmato).

Questo segmento è rimasto irrisolto perché FFmatch non è riuscito a convalidarlo con l'algoritmo di controllo che chiamo *sift check*. I motivi per cui il *sift check* è fallito possono essere molti: forse quei fotogrammi sono molto scuri, oppure la qualità di uno dei due è molto bassa, oppure quelle immagini sono di un tipo che non ha normalmente *keypoints*, oppure è fallito il controllo sulle proporzioni (si veda la sezione Problemi), ecc.

Diversamente dal caso precedente, qui non abbiamo una differenza interna. Infatti il campo *diff* della riga precedente coincide con quello della riga successiva ($\text{diff}=-592$).

Essendo un segmento di dimensioni molto contenute (24), lo possiamo collegare senza perderci tempo.

Ci sono 3 modi per chiudere velocemente una riga rossa di tipo $\{n-n\}$.

Il modo più rapido è quello di cliccare su menu Misc→ChiusuraFinale. Si tenga presente, però, che in questo modo verranno chiuse anche tutte le altre eventuali righe rosse di tipo $\{k-k\}$.

Un altro modo è quello di abilitare il menu Preferenze→Sequenze→ChiusuraFinale e di avviare il riconoscimento automatico (*F5*); alla fine del riconoscimento verrà fatta automaticamente la chiusura di tutte le righe rosse $\{n-n\}$.

Il terzo e ultimo modo è quello di fare il collegamento a mano, basta cliccare sul numero blu 127898, premere *Ctrl+Invio* e confermare a tutte le domande.

Nelle vecchie versioni di FFmatch l'opzione "Chiusura finale" era abilitata di default, quindi le righe rosse di tipo $\{n-n\}$ non capitavano mai.

Abilitare quell'opzione non è insensato, perché, se è vero che i controlli *sift check* non sono riusciti a stabilire l'equivalenza dei due segmenti, è tuttavia molto probabile che siano effettivamente equivalenti (cioè che una riga $\{n-n\}$ non contenga differenze).

Se però n è un valore molto alto (migliaia di fotogrammi), sarebbe opportuno che l'utente faccia un controllo visivo per assicurarsi che all'interno di quei grossi segmenti non vi siano differenze non rilevate. Per questa ragione nelle ultime versioni del programma quell'opzione è disabilitata di default: l'utente deciderà se abilitarla o meno e, nel secondo caso, potrà scegliere come gestire quel tipo di righe rosse.

Un altro caso speciale che merita di essere discusso è quello delle righe rosse del tipo $\{n-m\}$ dove n e m differiscono di 1 (in valore assoluto).

Come regola generale, vale per questi casi quanto già detto per le righe $\{n-n\}$: di solito si possono chiudere senza perderci troppo tempo. Non vale la pena cercare di allineare perfettamente le sequenze, individuando il fotogramma che una delle due ha e che manca all'altra: anche se dovessimo fare un collegamento "shiftato" di un fotogramma, l'eventuale asincronia audio/video (limitata a quel segmento) sarà al massimo di 40 ms e di solito non è percepibile.

Per chiudere velocemente una riga rossa del tipo {n-m} dove $|n-m|=1$, la procedura è leggermente più complicata di quella vista. Supponiamo di voler chiudere una riga rossa come la seguente:

```
[116410-127897] ---> [115818-127305] {011488} (diff=-592)
[127898-127921] -/-> ??? {24-23}
[127922-139446] ---> [127329-138853] {011525} (diff=-593)
```

La procedura da seguire è questa: si clicca sul numero blu 127898, si preme *Ctrl+Shift+freccia a destra*, si preme *Ctrl+Invio* e si conferma tutto.

In questo caso, il passaggio chiave è dato dalla combinazione di tasti *Ctrl+Shift+freccia a destra* che ha l'effetto di spostare le posizioni alla fine del segmento mantenendo la sincronia, quindi il primo filmato non si sposterà all'ultimo fotogramma (ventiquattresimo) del segmento, ma al penultimo (ventitreesimo), che verrà collegato all'ultimo (ventitreesimo) del secondo filmato. Dato che tale collegamento lascerà alla sua sinistra una coppia di segmenti di pari ampiezza (22 fotogrammi), il programma chiederà se si desidera collegare anche quelli: confermando faremo sparire tutta quella riga rossa. Il ventiquattresimo fotogramma del segmento del primo filmato resterà scollegato e finirà in fondo al report, tra i “frame(s) del primo filmato che non sono presenti nel secondo”.

Esiste un altro caso speciale che è una variante del precedente e tende a presentarsi quando uno dei due filmati ha subito un cambio “hard” di framerate: talvolta più differenze interne sono tali da compensarsi a vicenda; in altre parole, il campo *diff* delle righe verdi (alternate a righe rosse) prima cambia di valore di 1 (in valore assoluto), per poi tornare al valore precedente. Questa caratteristica può presentarsi molte volte in una stessa mappatura. In questi casi si può prendere in considerazione la possibilità di semplificare la mappatura rimuovendo un certo numero di righe verdi al fine di ottenere grossi segmenti rossi del tipo {n-n} (e quindi facilmente trattabili). La procedura per eliminare totalmente una riga verde è la seguente: si clicca su menu Pulizia→OttimizzaIDatiDiMapping e si conferma; poi si clicca sul numero blu più a sinistra della riga verde da rimuovere e si preme *Ctrl+Invio*: il programma chiede se si intende rimuovere il collegamento, si conferma e la riga verrà interamente rimossa. Se i dati di mapping non sono ottimizzati, la rimozione di un collegamento non comporterà necessariamente la rimozione totale della riga verde di cui il collegamento fa parte, anzi, più probabilmente verrà rimossa solo una

porzione di quella riga (dipende dalla storia passata, cioè dal modo in cui è venuta a crearsi quella riga verde).

Per fissare le idee su quanto appena detto, facciamo un esempio concreto. Supponiamo di avere una riga rossa del tipo {422-423}, seguita da una riga verde di ampiezza {000134}, seguita da una riga rossa del tipo {587-586}. Osserviamo che le due righe rosse si compensano, in quanto nella prima il primo filmato ha un fotogramma in meno del secondo e nella seconda ne ha uno in più. Molto probabilmente nelle due righe rosse non vi sono differenze significative (al di là del fatto che uno dei due filmati ha un fotogramma in più o in meno rispetto all'altro). In ogni caso basta fare un rapido controllo visivo (si vedano le indicazioni nella sezione Scorciatoie da tastiera). Se eliminiamo la riga verde (134 fotogrammi) con la procedura descritta sopra, otteniamo una unica riga rossa del tipo {1143-1143} ($422+134+587 = 423+134+586 = 1143$). Adesso la possiamo trattare con uno dei metodi già discussi per le righe rosse di tipo {n-n}.

Come accennato sopra, il caso delle differenze interne (con $|n-m|=1$) che si presentano in modo da compensarsi, tende a presentarsi quando uno dei due filmati ha subito un cambio "hard" di framerate. Se siamo proprio in questa situazione, allora la semplificazione che otteniamo eliminando righe verdi (al fine di ottenere ampie righe rosse di tipo {n-n}) non è un'approssimazione ma una correzione degli errori introdotti nella sequenza da una "decimazione" del frame rate.

In ogni caso, questo tipo di semplificazione ha almeno un altro vantaggio: come dovrebbe essere chiaro, ogni passaggio da una riga della mappatura alla riga successiva comporta una discontinuità nell'audio generato. Nella maggior parte delle situazioni, tali discontinuità si presentano tra una scena e l'altra del film, pertanto di solito sono difficilmente percepibili. Resta comunque il fatto che riducendo il numero di righe della mappatura si riducono le discontinuità, ed è un fatto positivo.

Per concludere questa sezione, resta da discutere il caso delle righe rosse di tipo {n-m} dove $|n-m|>1$.

Questo caso è il più impegnativo perché va gestito con cura, altrimenti in quel segmento avremo asincronie audio/video percepibili.

Riconsideriamo questa riga rossa:

```
[025590-027504] ---> [025038-026952] {001915} (diff=-552)
[027505-027834] -/-> ??? {330-327}
[027835-049744] ---> [027280-049189] {021910} (diff=-555)
```

In questo caso abbiamo $n=330$ e $m=327$, quindi $|n-m|=3$: in quel segmento non risolto, il primo filmato ha 3 fotogrammi in più del secondo e il programma non è riuscito a localizzarli con precisione.

È chiaro che dobbiamo localizzare con precisione i fotogrammi del primo filmato che il secondo non ha, altrimenti in quella sequenza, della durata di circa 13 secondi, avremo un'asincronia audio/video di circa 120 ms.

I fotogrammi critici potrebbero trovarsi in qualsiasi punto di quella sequenza composta da 330 fotogrammi.

Tuttavia alcune posizioni sono più probabili di altre, in particolare queste due:

- i fotogrammi in più (nell'esempio considerato sono nel primo filmato e sono 3) sono i primi della sequenza;
- i fotogrammi in più sono gli ultimi della sequenza.

Questi due casi estremi si possono trattare facilmente con gli strumenti del programma, in particolare con i pulsanti di match: basterà trovare un fotogramma della prima (o seconda) sequenza irrisolta che abbia una differenza percepibile dai fotogrammi che lo precedono e lo seguono, dopodiché si clicca più volte sul primo pulsante di match (o sul secondo pulsante di match, se si è optato per un fotogramma del secondo filmato). A ogni click del primo (secondo) pulsante di match, il programma posiziona il secondo (primo) filmato sui fotogrammi che corrispondono al corrente del primo (secondo) rispettando gli allineamenti più ovvi (allineamento a sinistra e allineamento a destra). Se uno di tali allineamenti è corretto, basterà usare il primo o l'ultimo pulsante blu (spostamento sincrono sugli estremi dell'intervallo: *Ctrl+Shift+freccia a sinistra o destra*), forzare il collegamento (*Ctrl+Invio*) e confermare a tutte le domande. Quale dei due spostamenti sincroni sugli estremi occorre usare? Dipende da qual è l'allineamento corretto: se siamo nel primo caso (i fotogrammi in più sono i primi), l'allineamento corretto è quello a destra, quindi occorre premere *Ctrl+Shift+freccia a sinistra*; se siamo nel secondo caso (i fotogrammi in più sono gli ultimi), l'allineamento corretto è quello a sinistra, quindi occorre premere *Ctrl+Shift+freccia a destra*. Se non si è sicuri si può procedere così: si preme *Ctrl+Shift+freccia a sinistra* e si guardano le posizioni correnti dei due filmati: se sono entrambe immediatamente successive alle ultime della precedente riga verde, allora l'allineamento corretto è a sinistra e dobbiamo

premere *Ctrl+Shift+freccia a destra* e poi *Ctrl+Invio*, perché i fotogrammi in più sono gli ultimi della sequenza e vanno lasciati scollegati (finiscono in fondo al report come “frame(s) presenti in un filmato ma non nell’altro”); se, invece, non sono entrambe successive alle ultime della precedente riga verde, vuol dire che i fotogrammi in più sono i primi della sequenza, quindi siamo già nelle posizioni giuste e basta premere *Ctrl+Invio* e confermare il collegamento dell’intero segmento (i fotogrammi in testa restano scollegati e finiscono in fondo al report come “frame(s) presenti...”).

Nel precedente paragrafo ho scritto: “basterà trovare un fotogramma della prima (o seconda) sequenza irrisolta che abbia una differenza percepibile dai fotogrammi che lo precedono e lo seguono”. Adesso spiego come ci si sposta velocemente su un fotogramma interno a una sequenza (spostamento svincolato). Supponiamo di voler posizionare il primo filmato a circa metà di questa riga rossa:

```
[025590-027504] ---> [025038-026952] {001915} (diff=-552)
[027505-027834] -/-> ??? {330-327}
[027835-049744] ---> [027280-049189] {021910} (diff=-555)
```

Vogliamo andare a circa metà, quindi non c’è bisogno di fare il calcolo esatto: vediamo che l’estremo dell’intervallo è circa 27500 e che l’ampiezza dell’intervallo è un po’ più di 300, quindi sommiamo 150 (la metà di 300) a 27500 ottenendo 27650. Digitiamo 27650 nel primo spin numerico e premiamo *Invio*: in questo modo abbiamo spostato il primo filmato a circa metà di quel segmento scollegato. Se non siamo sicuri di non avere fatto errori, premiamo *Ctrl+F1* in modo da verificare che la riga rossa venga messa in evidenza (se non è così, abbiamo sbagliato qualcosa). Se invece vogliamo posizionare il secondo filmato a circa metà del segmento scollegato, si procede in modo del tutto analogo, con l’unica differenza che è leggermente meno immediato vedere l’estremo inferiore, che è il numero successivo dell’estremo superiore della precedente riga verde; nel nostro esempio, l’estremo superiore della precedente riga verde è 26952, quindi l’estremo inferiore del segmento scollegato (secondo filmato) è 26953, che possiamo arrotondare a 27000, a cui aggiungiamo 150 ottenendo 27150. Dunque, digitiamo 27150 nel secondo spin numerico e premiamo *Invio*: in questo modo abbiamo spostato il secondo filmato a circa metà di quel segmento scollegato.

Il metodo appena visto per posizionarsi a circa metà di un segmento scollegato può essere usato per “spaccare” eventuali grossi blocchi rimasti scollegati. Se i segmenti scollegati sono molto grandi (migliaia di

fotogrammi) si può cercare di spaccarli posizionandosi a circa metà e usare i pulsanti con le frecce verdi per individuare un cambio di scena. A questo punto si può tentare con i pulsanti di match, ma se i segmenti sono molto grandi è probabile che nascondano altre differenze interne (che il programma non è riuscito a rilevare) quindi è improbabile che i pulsanti di match forniscano il fotogramma correttamente corrispondente. Niente vieta però di usare i pulsanti con le frecce verdi per entrambi i filmati, in modo da individuare lo stesso cambio di scena. Quando si utilizzano i pulsanti con le frecce verdi, può capitare che i cambi di scena individuati siano lontani dalla posizione corrente e c'è pertanto il rischio di uscire dalla riga rossa su cui stiamo lavorando; come al solito, questo lo possiamo verificare premendo *Ctrl+F1* e *Ctrl+F2*. I pulsanti con le frecce verdi consentono usualmente di trovare facilmente due cambi di scena (e quindi fotogrammi) corrispondenti; a quel punto forziamo il collegamento (*Ctrl+Invio*) e la grande sequenza verrà suddivisa in due righe rosse più piccole (se abbiamo trovato i cambi di scena a circa metà, le righe rosse avranno circa metà ampiezza di quella di partenza). Dopo avere spaccato una riga rossa, vale sempre la pena far ripartire l'analisi automatica (*F5*), perché a volte, grazie a questo aiuto da parte dell'utente, vengono stabiliti molti collegamenti che erano sfuggiti in precedenza.

Torniamo per l'ultima volta alla riga rossa {330-327}. Abbiamo discusso come si procede nei casi in cui i 3 fotogrammi in più siano i primi o gli ultimi della sequenza del primo filmato. Se non si verifica né l'uno né l'altro caso, i 3 fotogrammi si trovano all'interno della sequenza e questo è sicuramente il caso più difficile da trattare. In linea di massima conviene procedere più o meno con il metodo già visto per "spaccare" un segmento di grandi dimensioni. Ma se il segmento non è di grandi dimensioni, come in questo caso (330 fotogrammi), non ha senso saltare tra cambi di scena usando i pulsanti con le frecce verdi. Ha senso posizionarsi a circa metà segmento e usare il pulsante di match, ma se i fotogrammi proposti non sono precisamente corrispondenti occorrerà fare spostamenti sincroni sui filmati per individuare una coppia di fotogrammi corrispondenti; a quel punto si forza il collegamento (ed eventualmente si fa ripartire l'analisi automatica), dopodiché si ripete il procedimento sulle righe rosse (più piccole) che si creano in seguito al collegamento. I casi più difficili di fotogrammi interni in più (o in meno) sono dovuti alla cattiva qualità di uno dei due filmati: se si va a controllare con un player il filmato (la posizione in formato hh:mm:ss è indicata accanto agli spin numerici ed è utile proprio

per questo tipo di controlli), si noteranno in quel segmento discontinuità sia nel video che nell'audio.

Scorciatoie da tastiera

(15 ottobre 2023)

In questa sezione elencherò le scorciatoie da tastiera che semplificano l'utilizzo di FFmatch.

(Evito di includere in questo elenco quelle che sono indicate nel menu a tendina del programma.)

F5: avvia il riconoscimento automatico; equivale al pulsante con le rotelline.

Esc: interrompe il riconoscimento automatico; equivale al pulsante di stop.

Ctrl+Invio: equivale al pulsante di collegamento.

Ctrl+F1: questa combinazione di tasti serve a evidenziare la riga della mappatura in cui si trova l'attuale fotogramma del primo filmato. Può essere utile per capire se, in seguito a spostamenti, siamo usciti dal segmento su cui stavamo lavorando.

Ctrl+F2: analogo al precedente ma relativo al secondo filmato.

Tab: ha la stessa funzione dei *pulsanti di match*, con questa precisazione: se il widget corrente è lo spin numerico del primo filmato, equivale al primo pulsante di match; se il widget corrente è lo spin numerico del secondo filmato, equivale al secondo pulsante di match (con "widget corrente" intendo il widget che ha il focus).

tasto freccia su: se il widget corrente è lo spin numerico del primo filmato, aumenta di 1 la posizione del primo filmato; se il widget corrente è lo spin numerico del secondo filmato, aumenta di 1 la posizione del secondo filmato.

tasto freccia giù: se il widget corrente è lo spin numerico del primo filmato, decrementa di 1 la posizione del primo filmato; se il widget corrente è lo spin numerico del secondo filmato, decrementa di 1 la posizione del secondo filmato.

Shift+freccia su: se il widget corrente è lo spin numerico del primo filmato, aumenta di 10 la posizione del primo filmato; se il widget corrente è lo spin numerico del secondo filmato, aumenta di 10 la posizione del secondo filmato.

Shift+freccia giù: se il widget corrente è lo spin numerico del primo filmato, decrementa di 10 la posizione del primo filmato; se il widget corrente è lo spin numerico del secondo filmato, decrementa di 10 la posizione del secondo filmato.

Shift+freccia a sinistra: equivale al terzo pulsante blu (spostamento sincrono).

Shift+freccia a destra: equivale al quarto pulsante blu (spostamento sincrono).

Ctrl+freccia a sinistra: equivale al secondo pulsante blu (spostamento sincrono).

Ctrl+freccia a destra: equivale al quinto pulsante blu (spostamento sincrono).

Ctrl+Shift+freccia a sinistra: equivale al primo pulsante blu (spostamento sincrono).

Ctrl+Shift+freccia a destra: equivale al sesto pulsante blu (spostamento sincrono).

Oltre alle scorciatoie elencate, si consideri che, ogni volta che il programma pone una domanda all'utente, è possibile rispondere, invece di cliccare sul relativo pulsante, battendo i tasti S o Y in caso di risposta affermativa, oppure N in caso di risposta negativa. Inoltre il tasto Esc equivale ad annullare e tipicamente ha lo stesso significato di una risposta negativa.

La combinazione di tasti che è forse la più utile tra quelle elencate è *Ctrl+freccia a destra*, in quanto permette di controllare velocemente la sincronia (ed eventuali perdite di sincronia) di una sequenza, anche abbastanza lunga. La tecnica è la seguente: si clicca sul numero blu a sinistra della riga che vogliamo ispezionare; ciò ha l'effetto di posizionare i filmati all'inizio del segmento. Poi si preme la combinazione di tasti

Ctrl+freccia a destra e si continua a tenerla premuta, sempre controllando che il fotogramma mostrato di sopra resti in sincronia con quello mostrato di sotto. Se durante tale scansione si esce dalla riga attuale, il programma emetterà un breve suono.

Si osservi che, sebbene in linea di principio sia possibile effettuare un'analoga scansione all'indietro (*Ctrl+freccia a sinistra* oppure *Shift+freccia a sinistra*), in pratica conviene sempre fare scansioni in avanti, perché i formati di compressione video sono progettati per essere decodificati velocemente nel tempo che avanza, e la decodifica dei “fotogrammi precedenti all'attuale” è molto inefficiente (disponendo di molta memoria, si può pensare di aumentare il parametro Preferenze→Misc→FrameBufSize, ma la possibilità di tornare indietro velocemente resta sempre limitata a pochi fotogrammi).

L'entità dello spostamento sincrono ottenuto premendo *Ctrl+freccia a destra* ha il valore di default di 10 fotogrammi in avanti per volta. Questo valore può essere modificato nel menu Preferenze→Misc→SupStepIncr. Se il segmento che vogliamo ispezionare è lungo migliaia di fotogrammi, si può pensare di aumentare quel valore al fine di ridurre i tempi necessari alla scansione.

Imparare a usare bene le scorciatoie da tastiera è di fondamentale importanza, soprattutto *Ctrl+freccia a destra* come già detto, in combinazione con *Tab*, *Ctrl+Shift+freccia a sinistra*, *Ctrl+Shift+freccia a destra*, *Ctrl+Invio*. Acquisendo molta pratica con queste combinazioni di tasti, si possono risolvere abbastanza facilmente situazioni non banali. (A volte mi è capitato di fare dei mux senza usare il riconoscimento automatico, ma solo questi strumenti “a mano”).

Forzare collegamenti

(15 ottobre 2023)

In questa sezione spiego il significato del *pulsante di collegamento* che di base serve a forzare collegamenti (può essere attivato anche con la combinazione di tasti *Ctrl+Invio*).

Il pulsante di collegamento effettua il collegamento tra i due frame selezionati se e solo se valgono le seguenti condizioni:

- i due frame sono entrambi liberi;
- i due frame sono direttamente collegabili, cioè non esistono collegamenti stabiliti in precedenza che ostacolano il nuovo collegamento (il modello matematico su cui si basa il programma non consente collegamenti “incrociati”; questo limite può però essere aggirato tramite *spostamenti*, si veda la sezione [Spostamenti di sequenze](#)).

Se valgono le precedenti condizioni, il programma chiederà conferma all'utente e, in caso affermativo, collegherà i due frame. Inoltre, se la creazione di tale collegamento dovesse lasciare intervalli liberi di pari lunghezza, il programma chiederà se si desidera collegare anche quelli (questa caratteristica permette anche di collegare interi segmenti: basterà collegare prima due estremi e poi gli altri due).

Se le precedenti condizioni non valgono, il programma non collegherà i due frame nella attuale richiesta ma **consentirà di creare le condizioni affinché siano collegabili ripetendo la richiesta.**

(Le versioni più vecchie di FFmatch informavano l'utente dell'impossibilità di fare il collegamento. Le nuove versioni, invece, rendono sempre possibile il collegamento, eventualmente cancellando precedenti collegamenti oppure facendo uno spostamento.)

Si noti che se le precedenti condizioni non valgono, allora varrà esattamente una delle seguenti condizioni:

- almeno uno dei due frame è già collegato;
- i due frame sono entrambi liberi, ma non sono direttamente collegabili.

Nel primo caso, il programma propone all'utente la rimozione dei precedenti collegamenti che coinvolgono i due frame. Confermando, i collegamenti critici verranno rimossi e si potrà ripetere la richiesta di collegamento.

Nel secondo caso, il programma propone all'utente lo spostamento dell'intervallo irrisolto del secondo filmato di cui fa parte il secondo frame selezionato alla più vicina posizione che renda collegabili i due frame. Questo secondo caso merita una spiegazione approfondita, che verrà data nella prossima sezione.

Spostamenti di sequenze (15 ottobre 2023)

Nella sezione precedente abbiamo visto che, chiedendo di collegare due frame liberi ma non direttamente collegabili (cioè, tali da creare un “incrocio” con uno o più collegamenti stabiliti in precedenza), il programma propone all’utente lo spostamento di un intervallo.

Per fissare le idee, vediamo un esempio in tutti i dettagli.

Ho preparato una coppia di filmati, partendo da un film scelto a caso, splittandolo in cinque parti (A, B, C, D, E) e creando un secondo filmato ri assemblando le parti in ordine leggermente diverso: A+D+C+B+E.

Ho chiamato i due filmati ABCDE.avi e ADCBE.avi:

```
filmato 1: ABCDE.avi
risoluzione: 592 x 256
display aspect ratio: 37:16 = 2.313
framerate: 25 fps
durata: 01:44:07 (6247.2 s)
numero approssimato di fotogrammi: 156178
codec: mpeg4 (decoder: mpeg4)

filmato 2: ADCBE.avi
risoluzione: 592 x 256
display aspect ratio: 37:16 = 2.313
framerate: 25 fps
durata: 01:44:07 (6247.2 s)
numero approssimato di fotogrammi: 156178
codec: mpeg4 (decoder: mpeg4)
```

I due filmati hanno esattamente gli stessi frame, ma sono montati in ordine diverso.

Faccio partire l’analisi (tasto *F5*) e alla fine, la situazione è questa:

```
(1) ABCDE.avi
(2) ADCBE.avi

free (1): 63473/156178
free (2): 836/156178

controlli: 52 (passati: 52)
rapporto orizzontale: 1
rapporto verticale: 1
internal diff: 2
free segm: 1

[000001-031940] ---> [000001-031940] {031940} (diff=0)
[031941-067104] ---> [094578-129741] {035164} (diff=62637)
[067105-130577] -/-> ??? {63473-836}
[130578-156178] ---> [130578-156178] {025601} (diff=0)

frame(s) del secondo filmato che non sono presenti nel primo:
```



```
[031941-094577]2---> [031940] {62637}
```

Restano 836 frame liberi, facilmente collegabili con due opportune richieste al pulsante di collegamento, dopodiché ci troviamo in questa situazione:

```
(1) ABCDE.avi
(2) ADCBE.avi

free (1): 0/156178
free (2): 0/156178

controlli: 52 (passati: 52)
rapporto orizzontale: 1
rapporto verticale: 1
internal diff: 2
free segm: 0

[000001-031940] ---> [000001-031940] {031940} (diff=0)
[031941-067604] ---> [094578-130241] {035664} (diff=62637)
[130242-156178] ---> [130242-156178] {025937} (diff=0)

frame(s) del primo filmato che non sono presenti nel secondo:
[067605-130241]1---> [130241] {62637}

frame(s) del secondo filmato che non sono presenti nel primo:
[031941-094577]2---> [031940] {62637}
```

Osserviamo che il programma ha collegato la parte A (i primi 31940 frame di entrambi i filmati), la parte E (gli ultimi 25937 frame) e la parte B (che nel primo filmato segue la parte A, mentre nel secondo è spostata dopo la parte C).

Restano libere le parti C e D che nel primo filmato sono nell'ordine C+D, mentre nel secondo sono nell'ordine inverso D+C.

La situazione è dunque questa: **ABCDE** vs **ADCBE**.

Per quanto riguarda le versioni più vecchie di FFmatch, la storia finiva qui, perché la strada imboccata non consentiva di proseguire oltre.

Osserviamo che il programma avrebbe potuto anche trovare il collegamento della parte C (o della parte D), prima di quello della parte B, e in tal caso avremmo ottenuto un risultato diverso.

Quello che ci serve adesso è esattamente uno spostamento.

Le righe che vengono mostrate in fondo al report come “frame che sono presenti in uno ma non nell'altro” non sono altro che “righe rosse” in cui uno dei due segmenti è vuoto.

In questo caso abbiamo due righe rosse in cui, rispettivamente, 62637 frame sono presenti nell'uno ma non nell'altro. Si tratta ovviamente delle parti C e D.

Ci basterà chiedere il collegamento di un qualsiasi frame di queste due zone.

Ad esempio, clicco sul numero blu “067605”, e sul secondo spin ottengo “130241”. Al posto di “130241” digito “31941” e batto *Invio*.

Adesso i frame correnti sono “67605” e “31941”, rispettivamente.

Non sono frame corrispondenti, ed è ovvio, perché sappiamo che 67605 (del primo filmato) è il primo del blocco C+D (e quindi il primo di C), mentre 31941 (del secondo filmato) è il primo del blocco D+C (e quindi il primo di D).

Ma non ci interessa che siano corrispondenti, ci serve soltanto che siano negli intervalli irrisolti che vogliamo rendere potenzialmente collegabili.

Chiediamo quindi il collegamento e il programma ci propone:

```
Vuoi spostare [31941-94577] (filmato 2) verso 130242?
```

Confermiamo e otteniamo questa situazione:

```
(1) ABCDE.avi
(2) ADCBE.avi

free (1): 62637/156178
free (2): 62637/156178

controlli: 52 (passati: 52)
rapporto orizzontale: 1
rapporto verticale: 1
internal diff: 0
free segm: 1

[000001-067604] ---> [000001-067604] {067604} (diff=0)
[067605-130241] -/-> ??? {62637-62637}
[130242-156178] ---> [130242-156178] {025937} (diff=0)
```

Come si vede, dopo lo spostamento, gli intervalli irrisolti sono tornati in gioco. Prima avevamo 3 righe verdi, adesso sono diventate due perché lo spostamento le ha rese adiacenti e quindi accorpabili; per capire questo punto, basta considerare che la situazione precedente allo spostamento era **ABCDE** vs **ADCBE**, mentre dopo lo spostamento è diventata **ABCDE** vs **ABDCE**: le parti A e B sono diventate un tutt'uno.

A questo punto facciamo partire l'analisi (F5) e finiamo in questa situazione:

```
(1) ABCDE.avi
(2) ADCBE.avi

free (1): 0/156178
free (2): 0/156178

controlli: 56 (passati: 56)
rapporto orizzontale: 1
rapporto verticale: 1
```

```

internal diff: 2
free segm: 0

[000001-067604] ---> [000001-067604] {067604} (diff=0)
[067605-097843] ---> [100003-130241] {030239} (diff=32398)
[130242-156178] ---> [130242-156178] {025937} (diff=0)

frame(s) del primo filmato che non sono presenti nel secondo:
[097844-130241]1---> [130241] {32398}

frame(s) del secondo filmato che non sono presenti nel primo:
[067605-100002]2---> [067604] {32398}

```

Constatiamo che il programma ha trovato il collegamento della parte C e quindi ci troviamo adesso in questa situazione: **ABCDE** vs **ABDCE**.

Analogamente a quanto già visto in precedenza, la parte D non è direttamente collegabile (la strada è sbarrata dal collegamento della parte C) e finisce in fondo al report.

Non ci resta che chiedere di spostare anche la parte D: clicco sul numero blu “097844”, sul secondo spin ottengo “130241” che cambio con “67605” e batto *Invio*.

Adesso vedo che i frame sopra e sotto sono effettivamente identici: si tratta evidentemente del primo frame della parte D.

Premiamo *Ctrl+Invio* e il programma ci chiede:

Vuoi spostare [67605-100002] (filmato 2) verso 130242?

Confermiamo e otteniamo questa situazione:

```

(1) ABCDE.avi
(2) ADCBE.avi

free (1): 32398/156178
free (2): 32398/156178

controlli: 56 (passati: 56)
rapporto orizzontale: 1
rapporto verticale: 1
internal diff: 0
free segm: 1

[000001-097843] ---> [000001-097843] {097843} (diff=0)
[097844-130241] -/-> ??? {32398-32398}
[130242-156178] ---> [130242-156178] {025937} (diff=0)

```

In altri termini, siamo giunti a questa situazione: **ABCDE** vs **ABCDE**.

Osserviamo che le parti A+B+C costituiscono adesso un unico segmento di 97843 frame.

Premiamo *F5* e otteniamo la situazione finale:

```

(1) ABCDE.avi
(2) ADCBE.avi

free (1): 0/156178

```

```
free (2): 0/156178
controlli: 60 (passati: 60)
rapporto orizzontale: 1
rapporto verticale: 1
internal diff: 0
free segm: 0

[000001-156178] ---> [000001-156178] {156178} (diff=0)
```

Abbiamo così ottenuto la mappatura completa tra i due filmati e abbiamo quindi finito.

Vale la pena fare un paio di osservazioni. La prima è che, ad ogni spostamento, la numerazione dei frame del secondo filmato cambia. La seconda è che il “timer” alla destra del secondo spin non è più strettamente crescente (linearmente al crescere del numero di frame nello spin) ma indica la posizione hh:mm:ss relativa al frame corrente.

Non ci sono limiti al numero di spostamenti che si possono fare.

Forse in futuro aggiungerò la possibilità che il programma cerchi da solo gli spostamenti potenzialmente fruttuosi.

Preferenze

In questa sezione viene spiegato il significato dei parametri modificabili dalla finestra delle preferenze.

Sequenze

Init thres./avg: [fixme]

Min thres./avg: [fixme]

Thres. ratio: [fixme]

Min int. blks: [fixme]

Min avg blk size: [fixme]

Chiusura finale: se abilitata, alla fine di ogni sessione di riconoscimento automatico eventuali righe rosse del tipo {n-n} verranno automaticamente collegate (si veda la sezione [Approfondimenti](#) per una spiegazione più dettagliata).

Controlli

Checks per interval+extra: è il numero di controlli (“sift check”) da effettuare per ogni coppia di segmenti potenzialmente corrispondenti (e quindi collegabili); il primo valore indica il numero di controlli nel caso standard in cui la coppia di segmenti è stata trovata tramite pattern matching sulle sequenze di lunghezze di scene. Il secondo valore indica un numero di controlli aggiuntivi (al valore precedente) da fare nei casi particolari in cui la coppia di segmenti è ipotizzata in base a qualche criterio (allineamenti a collegamenti già stabiliti in precedenza e simili). I valori di default (4 e 8) significano che nei casi standard vengono fatti 4 controlli, mentre negli altri casi ne vengono fatti 12 (4+8). Si tenga presente che una coppia di sequenze individuata tramite pattern matching ha un’alta probabilità di essere effettivamente corrispondente, mentre negli altri casi la probabilità è nettamente più bassa: il programma arriva a generare

ipotesi soprattutto in presenza di differenze interne, e in tali casi un numero maggiore di controlli è utile sia per convalidare l'ipotesi sia per discriminare tra ipotesi rivali; in genere il programma fa pochi (o nemmeno uno) collegamenti errati e questi sono quasi sempre dovuti a una errata discriminazione tra ipotesi rivali, quindi può avere senso aumentare il valore del secondo parametro.

Min keypoints+min ratios: questi tre parametri sono usati soltanto quando vengono tentate strade alternative alla ricerca di schemi di pattern matching tra lunghezze di scene; il primo parametro (“min keypoints”) indica il minimo numero di keypoints che devono avere le immagini da controllare, gli altri due (“min ratio” e “min ratio lum”) sono soglie che determinano come vadano gestiti i controlli nel caso in cui le immagini hanno abbastanza keypoints. Può avere senso provare a modificare un po' questi parametri solo in casi estremi in cui si notano tanti messaggi del tipo “delayed (dlum=...)”, altrimenti è meglio lasciare i valori di default.

Min agreements: il primo collegamento tra segmenti stabilito dal programma è il più critico, perché i successivi possono beneficiare dei dati relativi ai precedenti; per compensare l'assenza di informazioni, i criteri per superare il controllo sono più severi all'inizio del riconoscimento automatico. È per questo che il programma, di solito, impiega un po' più di tempo per fare il primo collegamento. Questo parametro stabilisce il minimo numero di linee rette concordanti (linearmente coerenti) affinché il controllo sia statisticamente significativo. Il valore di default (12000) sembra abbastanza severo da garantire un'alta probabilità che il primo collegamento sia corretto. In casi particolari può succedere, però, che a causa di una qualità troppo bassa di uno dei filmati questo valore sia troppo alto e il programma non riesca mai a trovare una prima coppia potenzialmente collegabile di segmenti che superi il controllo. Tali casi sono caratterizzati dal fatto che il programma effettua tanti controlli ma questi vengano rimandati (messaggio “delayed”). In questi casi si può provare a interrompere l'analisi (*Esc*), abbassare questo valore e far ripartire l'analisi (*F5*). Se, abbassando questo valore, il programma riesce a fare i primi collegamenti, è consigliabile interrompere l'analisi e controllare la correttezza di tali collegamenti (abbassando questo valore, aumenta la probabilità di errore nei primi collegamenti). Se sono corretti, si può far riprendere l'analisi.

Proport. thres.: il significato di questo parametro è spiegato nella sottosezione “Proportionality check failed”.

Scalatura: se non è abilitata, i controlli (SIFT check) verranno effettuati sulle immagini così come sono state ottenute dagli stream. In genere questa non è una buona idea, specialmente se uno dei filmati ha una risoluzione alta (verranno generati troppi keypoints). Se l'opzione è abilitata, le immagini (potenzialmente corrispondenti) su cui effettuare il controllo verranno ridimensionate in modo da mantenere l'aspect ratio e da avere all'incirca il numero di pixel scelto (il valore di default è 250000). Aumentare quel valore (numero di pixel) ha l'effetto di aumentare il numero di keypoints rilevati. Questo può essere utile se in almeno uno dei filmati non vengono rilevati abbastanza keypoints. Ad esempio, se si nota che il programma effettua molti controlli (e quindi individua corrispondenze tra sequenze), ma questi falliscono per carenza di keypoints (messaggio “too few keypoints”), può essere una buona idea interrompere l'analisi (*Esc*), raddoppiare questo valore (scalatura a un'area di 500000 pixel), cancellare i controlli falliti e far ripartire l'analisi (*F5*).

SIFT

Match thres.+min ratio: il primo di questi due parametri è una soglia che determina il criterio con cui un keypoint di un'immagine viene considerato equivalente a un keypoint dell'altra immagine. Senza entrare troppo nei dettagli tecnici, basti sapere che, aumentando quel parametro, tende ad aumentare anche il numero di linee rette che vengono disegnate tra l'immagine di sopra e quella di sotto (“matches” tra coppie di keypoint). Non è detto che sia una cosa positiva, anzi, di solito non lo è, perché le linee “extra” tendono a essere linearmente incoerenti con le altre. Il valore ottimale di “match thres.” è quello per cui si ottiene il massimo numero di linee linearmente coerenti e il minor numero di linee incoerenti con le prime. Il valore 0.68 dovrebbe essere buono per l'implementazione SIFT usata nelle ultime versioni del programma. Il secondo parametro (“min ratio”) è un'altra soglia che determina il criterio con cui i “matches” trovati sono considerati in numero insufficiente perché il controllo abbia senso (messaggio “too few matches”).

Filtri

I filtri servono più che altro a modificare il modo in cui i fotogrammi vengono visualizzati nella parte sinistra della finestra.

Il filtro più utile è “Gamma”, serve soprattutto quando un filmato è troppo scuro: scegliendo un valore superiore a 1, al filmato corrispondente verrà applicata una correzione che consentirà di vedere meglio alcuni dettagli.

Un altro filtro che occasionalmente può essere utile è “Aspect ratio”. Nei casi in cui le immagini mostrate nella parte sinistra della finestra dovessero apparire deformate (troppo alte e strette oppure troppo basse e larghe), si può rimediare disabilitando la rilevazione automatica dell’aspect ratio e digitando il valore corretto.

Misc

Frame buf size: è la dimensione dei buffer di lettura, espressa in numero di frames. Se si ha molta RAM libera, si può pensare di aumentare questo valore, senza esagerare. I buffer sono utili per velocizzare spostamenti all’indietro di poche posizioni; comunque, se si desidera fare scansioni manuali di segmenti di filmati, vale sempre il consiglio di cercare di scandire in avanti, per quanto possibile.

SCD threads: è il numero di threads da usare nella fase di scansione dei filmati. Per trovare il numero ottimale per il proprio sistema si può procedere così: si imposta a 1 questo valore, si cancellano i dati “scd” dal menu Pulizia (se necessario), si preme *F5* e si controlla la velocità di lettura frames a regime; poi si interrompe la scansione (*Esc*), si aumenta di 1 questo valore e si ripete la procedura; il valore ottimale è l’ultimo per cui si è verificato un aumento di velocità apprezzabile. (Nota: il valore ottimale dovrebbe essere compreso tra 1 e il valore CPUs mostrato nel menu Aiuto→Info).

Disegno linee: se abilitato, durante i controlli, nella parte sinistra della finestra vengono tracciate linee rette tra coppie di keypoints dei due filmati che vengono considerate corrispondenti.

Sup. step. incr.: è l’entità (numero di frames) dello spostamento sincrono della seconda e della quinta freccia blu (*Ctrl+freccia a destra* e *Ctrl+freccia a*

sinistra).

Debug: meglio lasciarlo disabilitato.

Suono: se abilitato, il programma emetterà alcuni suoni per segnalare determinati eventi; è utile soprattutto durante una sincronizzazione manuale per segnalare ogni passaggio da un segmento a un altro (ulteriori dettagli si trovano nella sezione Scorciatoie da tastiera).

CUVID: opzione sperimentale che non posso testare perché non ho una scheda NVIDIA. In generale, è meglio lasciarlo disabilitato. Chi volesse testarlo, innanzitutto deve avere una scheda NVIDIA, poi deve controllare che nel menu Aiuto→Info→Lavc tra i Decoders siano elencati h264_cuvid e hevc_cuvid; se si sta usando i precompilati Windows, sicuramente non ci sono, tuttavia è possibile sostituire le dll il cui nome inizia con "av" con omonime dll che sono state compilate con il supporto cuvid (si trovano facilmente già compilate in rete oppure è facile compilarle in ambiente mingw-w64 dal codice sorgente di ffmpeg), poi si controlla che nel menu Aiuto→Info→Lavc siano elencati h264_cuvid e hevc_cuvid; a quel punto si può abilitare questa opzione ed è consigliabile impostare a 1 il parametro "SCD threads". L'opzione ha effetto a partire dalle successive aperture dei filmati, quindi occorre riaprirli (*F1*, *F2*). Alcuni utenti mi hanno segnalato che c'è qualcosa che non va nell'ordine dei frame ottenuti con questo metodo, pertanto è consigliabile controllare prima di usarlo. Per controllare, si apra un filmato H.264 o H.265 con *F1* quando questa opzione è disabilitata; poi si abiliti l'opzione e si apra lo stesso filmato con *F2*. In questo modo il filmato verrà decodificato sia con il decoder software (sopra) che con il decoder hardware (sotto); cliccando su Info→StampaINomiDeiFilmati viene indicato il decoder usato: nel primo deve venire h264 o hevc e nel secondo deve venire h264_cuvid o hevc_cuvid. Poi ci si sposti in una posizione a caso del primo filmato e si inserisca la stessa posizione nel secondo spin: i frame sopra e sotto devono essere uguali. Poi si avanzi sia il primo che il secondo filmato per una decina di posizioni. Se sono sempre uguali, è tutto ok. Purtroppo, come già scritto, non ho una scheda per fare questi test.

Lingua: lingua da usare nella gui.

Problemi

In questa sezione vengono discussi i principali problemi che possono presentarsi nell'uso del programma.

Cambi “hard” di framerate

Dopo la scansione di entrambi i filmati, può capitare che il programma non riesca a stabilire nemmeno un collegamento. Di solito il primo collegamento non viene stabilito immediatamente ma dopo alcuni secondi, tuttavia, se dopo alcuni minuti di tentativi nel testo di sopra non è comparsa nemmeno una riga verde, allora è inutile lasciarlo in esecuzione: per qualche motivo, il programma non trova corrispondenze significative nelle sequenze delle lunghezze delle scene, pertanto conviene interrompere l'analisi (*Esc*) e analizzare il problema.

I motivi per cui il programma non trova corrispondenze sono tanti, ma quello che capita più spesso è il seguente: almeno uno dei due filmati ha subito un cambio “hard” di framerate.

Si tenga presente, comunque, che il programma potrebbe non trovare corrispondenze per altri motivi (vedi la successiva sottosezione), quindi la prima cosa da fare è capire la natura del problema dal comportamento del programma: se il riconoscimento automatico sembra “girare a vuoto” senza mai tentare un collegamento tra intervalli, allora è molto probabile che siamo in presenza di cambio hard di framerate. Se, invece, si nota che vengono tentati collegamenti tra intervalli, ma per qualche motivo i controlli non vanno a buon fine, allora è improbabile che si tratti di cambio hard, ma di altro problema; in quest'ultimo caso si veda la successiva sottosezione.

In questa sottosezione discuto il caso dei cambi hard e come possa essere in qualche modo trattato.

Innanzitutto devo spiegare cosa intendo con “cambio hard di framerate”.

I filmati che vengono dati in pasto a FFmpeg sono quasi sempre versioni digitali di filmati che in origine furono creati su pellicola, e quasi sempre i filmati originali consistono di sequenze di 24 fotogrammi al secondo. Vale la pena notare che lo storico standard televisivo PAL prevede una sequenza di

50 semiquadri al secondo (cioè 25 quadri al secondo) e il metodo storicamente più utilizzato per trasmettere un film in televisione (o per riversarlo su nastro VHS) è quello di accelerarne semplicemente la riproduzione, il che vuol dire che da ogni fotogramma della pellicola vengono ottenuti due semiquadri. Chiaramente questo metodo ha come effetto collaterale una riduzione della durata del film, perché si passa da 24 a 25 fps. Per fare un esempio, se quando ero giovane vedevo un film al cinema che durava 100 minuti e dopo qualche anno rivedevo lo stesso film in tv (o in VHS) notando che la durata era diventata di 96 minuti, mi chiedevo: avranno rimosso alcune scene? La risposta a questa domanda è: “no, il film c'è tutto ma è stato accelerato, quindi la durata originale di 100 minuti va moltiplicata per $24/25$ e la durata televisiva è $100 * 24/25 = 96$ minuti”. Di solito le persone non sono infastidite (e nemmeno se ne accorgono) di un cambio di framerate di questa entità ($24/25$). Il vantaggio di questo metodo è che lascia inalterata la sequenza originale dei fotogrammi su pellicola. Un cambio di framerate di questo tipo lo chiamo “soft” ed è facilmente ottenibile con i programmi di muxing più diffusi: basta remuxare un filmato modificando il framerate nominale; un tale remux non comporta alcuna ricodifica dello stream video, ma si limita a modificare il framerate nominale dichiarato nel contenitore e al ricalcolo dei timestamps dei frame. (Ovviamente occorrerà trattare opportunamente anche l'audio, ma è un dettaglio che in questo contesto non ci interessa.)

Un cambio di framerate che non è di tipo “soft” lo chiamo “hard” ed è caratterizzato dal fatto che altera in qualche modo l'originale sequenza di fotogrammi.

Uno dei principali responsabili di cambio hard è lo standard televisivo americano NTSC, che prevede una velocità di riproduzione pari a $30000/1001$ fps (circa 29,97 fps). In questo contesto non ci interessano i dettagli delle tecniche “telecine” usate per convertire un classico film a 24 fotogrammi al secondo in un segnale NTSC. È però interessante osservare che tali tecniche prevedono un preliminare cambio soft di framerate (“pulldown”) da 24 a $24000/1001$ fps (circa 23,976 fps) ed è facile imbattersi in filmati con questo framerate perché molti DVD NTSC contengono stream (MPEG-2) che preservano la sequenza originale dei fotogrammi e danno la possibilità ai decoder di applicare il telecine al momento della decompressione (ottenendo uno stream a 29,97 fps) oppure di riottenere appunto la sequenza originale (a 23,976 fps); si noti che, nel secondo caso, si tratta nient'altro che di un cambio soft (da 24 a 23,976 fps) che ha senso mantenere solo per coerenza con l'audio (del DVD). Inoltre, ci basti sapere

che in linea di principio una conversione “telecine” da 24 fps (pellicola originale) a 29,97 fps sarebbe reversibile, cioè sarebbe possibile riottenere da un filmato a 29,97 fps l’esatta sequenza dei fotogrammi della pellicola originale, applicando specifici algoritmi di “inversione di telecine”. Ma in pratica questo può essere fatto solo in determinati casi, mentre in altri la sequenza originale va perduta a causa di filtraggi successivi al telecine.

Fatte queste premesse, esiste un’ampia casistica di conversioni hard, che a mio avviso non sono mai giustificate. Forse il motivo per una conversione hard che abbia un minimo di senso è quello per cui viene data priorità a due esigenze: ottenere un filmato che abbia un ben preciso framerate e non voler modificare una traccia audio. Per i miei gusti, però, preservare l’integrità della sequenza originale dei fotogrammi viene prima di tutto. Si consideri, inoltre, che l’esigenza di ottenere un filmato con un ben preciso framerate (tipicamente 25 fps) poteva avere senso quando il filmato andava riprodotto su un vecchio televisore, mentre oggi non vedo altre ragioni valide.

Torniamo alla mia definizione di cambio “hard”: si tratta di un cambio che altera in qualche modo l’originale sequenza di fotogrammi. In quali modi la altera? Come accennato, esiste un’ampia casistica ed è impossibile fare un elenco completo dei casi. Esistono però due metodi più ovvi di cambio hard: rimuovere frame e duplicare frame. Entrambi questi metodi compromettono la fluidità della riproduzione. Eliminare frame equivale a prendere una pellicola e tagliarla ogni circa 20 o 25 fotogrammi, poi tagliare un fotogramma all’estremo di ogni striscia e buttarlo via, e infine giuntare tutti i pezzetti per ottenere una nuova pellicola. Il problema è che la pellicola ottenuta è irrimediabilmente mutilata. Può sembrare strano, ma una percentuale non trascurabile di filmati che circolano in rete ha subito questo barbaro trattamento. Probabilmente molti di questi casi succedono involontariamente per una conversione di framerate fatta male. Comunque sia, il punto è che un filmato ottenuto rimuovendo o duplicando frame perde la sua naturale fluidità nella riproduzione: è vero che nelle scene poco movimentate la perdita di fluidità è poco (o niente) percepibile, ma le scene più movimentate procedono “a scatti” e non potrebbe essere altrimenti. Oltre alla rimozione e duplicazione di frame, esistono molti altri casi di cambi hard, che consistono essenzialmente nella creazione di frame “ibridi” che non corrispondono più a un unico fotogramma della pellicola originale ma a una qualche combinazione di più fotogrammi. Il caso più tipico è proprio quello, già menzionato, di filmato a 29,97 fps di provenienza cinematografica, che viene ricodificato (magari più volte e subendo filtraggi

vari a più riprese) proprio a 29,97 fps; in questo tipo di filmato, per mantenere la durata originale (in realtà, una durata leggermente più alta di quella originale per via del pulldown da 24 a 24000/1001) non vengono duplicati frame, ma il numero di frame viene aumentato combinando i frame originali secondo un determinato schema (da 4 frame se ne ottengono 5, in modo da passare da 24000/1001 a 30000/1001). Al di là dei dettagli, ciò che ci interessa sapere è che in un filmato a 29,97 fps che proviene da una pellicola a 24 fps i frame non trovano una corrispondenza con i fotogrammi della pellicola.

FFmatch è in grado di rilevare probabili cambi hard soltanto in situazioni molto particolari: in pratica quando i framerate dei filmati differiscono in modo significativo (il tipico caso è quello in cui un filmato ha un framerate compreso tra 23,976 e 25 fps e l'altro è a 29,97 fps). Quando il programma rileva questa situazione, nel testo di sotto viene stampato un avvertimento di colore rosso che può essere utile all'utente allo scopo di non perdere tempo a fare l'analisi automatica. Ma in generale non è possibile rilevare un cambio hard basandosi solo sui framerate e sulle durate. Per fare un esempio, mi sono capitati molti filmati a 25 fps che avevano subito un cambio hard; si può pensare di riconoscere questi casi confrontando il numero stimato dei frame dei due filmati, ma le differenze nel numero dei frame possono essere dovute alla presenza/assenza di scene nelle due versioni del film, quindi questo semplice criterio non è applicabile. Per sapere come stanno le cose, il programma dovrebbe accorgersi che non sta trovando nessuna corrispondenza nelle sequenze delle lunghezze delle scene, dopodiché dovrebbe provare a "scalare" le lunghezze di una delle due sequenze moltiplicandole per un fattore di scala "plausibile" e verificare se tale modifica consente di trovare corrispondenze. Ma quali sono i fattori di scala "plausibili"? Se uno dei due filmati ha framerate 29,97 fps, allora il fattore di scala più plausibile è 4/5 perché, come abbiamo detto in precedenza, una sequenza a 29,97 fps è probabilmente ottenuta dalla sequenza originale applicando uno schema che sostituisce 4 fotogrammi con 5 frame. Questo è l'unico caso in cui si possa ipotizzare un fattore di scala plausibile, in tutti gli altri casi occorre procedere per tentativi. Sicuramente si può tentare di scrivere algoritmi per automatizzare questo tipo di analisi, però è molto impegnativo, non credo di volerlo fare. Inoltre, di solito l'utente può capire velocemente come stanno le cose, con il metodo che spiego di seguito.

All'inizio di questa sottosezione ho scritto: "se dopo alcuni minuti di tentativi nel testo di sopra non è comparsa nemmeno una riga verde, allora

è inutile lasciarlo in esecuzione [...] conviene interrompere l'analisi (*Esc*) e analizzare il problema". Il modo più ovvio per analizzare il problema è quello di usare i pulsanti con le frecce verdi per individuare uno stesso cambio di scena. Se i filmati sono versioni dello stesso film, individuare uno stesso cambio di scena è di solito molto facile, soprattutto con i pulsanti con le frecce verdi. A questo punto ci si sposta in sincrono in avanti, un frame per volta (basta premere *Shift+frecchia a destra*) e si controlla visivamente che i frame mostrati (sopra e sotto) siano sempre corrispondenti. Se almeno uno dei due filmati ha subito un cambio hard, dopo un certo numero di posizioni (sempre meno di 30) deve succedere qualcosa di strano: innanzitutto il sincronismo viene perso, poi si presenta uno dei seguenti casi: uno dei due filmati ha saltato un frame (e quindi si è portato avanti di un fotogramma rispetto all'altro); oppure uno dei due filmati ha un frame duplicato cioè uguale al precedente (quindi è rimasto indietro rispetto all'altro); oppure uno dei due filmati ha un frame che è una combinazione pasticciata di più fotogrammi. In tutti questi casi è possibile fare una stima del fattore di scala da applicare al filmato che ha subito cambio hard, ad esempio calcolando il rapporto tra le differenze di due posizioni corrispondenti (vanno scelte non troppo vicine e tali da non contenere differenze nei frame che non siano quelle dovute al cambio hard). Ad esempio, se uno dei due filmati è a 29,97 fps e l'altro non ha subito cambi hard, dovremmo scoprire facilmente che il sincronismo viene perso molto rapidamente e che il fattore di scala è 4/5.

Quando abbiamo un'idea abbastanza precisa del fattore di scala corrispondente al cambio hard, possiamo creare una copia "dummy" da usare al suo posto.

A titolo di esempio, se il filmato che ha subito cambio hard è a 29,97 fps e il fattore di scala è 4/5, possiamo ottenere velocemente una copia dummy con un comando simile a questo:

```
mencoder "input-path" -vf harddup -ovc xvid -xvidencopts fixed_quant=2 -nosound -mc 0 -o dummy.avi -ofps 24000:1001
```

Il comando sopra genera il file `dummy.avi` che avrà framerate 23,976 fps e avrà la stessa identica durata del filmato in input. Si tratta chiaramente di un cambio hard che serve a compensare il cambio hard subito dal filmato a 29,97 fps (nella creazione dello stream verrà scartato un frame ogni gruppo di 5). Se non abbiamo fatto errori, il filmato ottenuto potrà essere usato in `FFmatch` al posto di quello da 29,97 fps e stavolta dovrebbero essere rilevate molte corrispondenze. Come accennato nella sezione [Approfondimenti](#), è

abbastanza probabile che la mappatura ottenuta quando un filmato ha subito un cambio hard (come in questo caso) presenti righe rosse di tipo $\{n-m\}$ con $|n-m|=1$ che tendono a compensarsi. Rimando a quella sezione per i consigli su come conviene procedere.

Più in generale, se il filmato che ha subito cambio hard ha framerate pari a x fps e abbiamo stimato un fattore di scala pari a k , useremo un comando come quello sopra in cui però specifichiamo l'opzione “-ofps y ” dove $y=x*k$. (Nell'esempio avevamo detto che il filmato era a 29,97 fps e il fattore di scala $4/5$, quindi $y=(30000/1001)*(4/5)=24000/1001$).

Il metodo descritto in questa sottosezione può sembrare un po' complesso, ma in fondo è abbastanza praticabile. Personalmente l'ho applicato molte volte: di solito si ottengono mappature meno soddisfacenti rispetto a quando si parte da coppie di filmati che non hanno subito cambi hard, ma sono comunque sistemabili (di nuovo, rimando alle prime sezioni della guida). Ringrazio ErTenebra perché, molti anni fa, fu il primo a comunicarmi di essere riuscito a usare il programma dopo avere applicato una “decimazione” a un filmato a 29,97 fps.

Difficoltà a stabilire i primi collegamenti

Il riconoscimento automatico può fallire anche perché non riesce a stabilire i primi collegamenti, pur trovando corrispondenze tra sequenze di lunghezze di scene.

I primi collegamenti sono più critici dei successivi, non potendo contare sulle informazioni ricavabili dai collegamenti precedenti.

In molti casi il riconoscimento automatico impiega un po' più di tempo per stabilire i primi collegamenti, ma nel giro di una manciata di secondi lo scoglio iniziale viene superato e la generazione dei collegamenti procede senza ulteriori intoppi.

A volte questo non succede, perché uno dei filmati è di qualità troppo bassa o comunque ha particolari caratteristiche che ostacolano i controlli basati su corrispondenze tra keypoints.

Si può tentare di trattare tali casi con un procedimento di questo tipo: si interrompe l'analisi infruttuosa (*Esc*), si modificano alcuni parametri nelle preferenze (*Ctrl+P*) e si fa ripartire l'analisi (*F5*).

Le modifiche da apportare sono le seguenti: aumentare il numero di pixel del parametro “Scalatura” (il numero di keypoints delle immagini da controllare tenderà ad aumentare) e abbassare il valore del parametro “Min agreements”. Per ulteriori dettagli si veda la sezione [Preferenze](#).

“Proportionality check failed”

Tipicamente uno dei due filmati è scalato rispetto all’altro con un fattore di scala costante. Più precisamente, i fattori di scala sono due: orizzontale e verticale. Tali fattori vengono mostrati da FFmatch nel testo di sopra in questi due campi:

```
rapporto orizzontale: ...  
rapporto verticale: ...
```

I numeri indicati sono i fattori di scala medi per passare dal primo al secondo filmato.

Esistono casi in cui il fattore di scala non è costante, cioè ci sono alcune scene che in uno dei due filmati hanno subito una scalatura diversa rispetto al resto del film (e anche un diverso cropping).

Il controllo sul fattore di scala (Proport.thresh. = 0.9), di solito è utile per prevenire eventuali collegamenti errati. Il valore 0,9 significa che siamo disposti a tollerare uno scostamento del 10% dall’attuale fattore di scala medio.

Nei casi in cui il fattore di scala non è costante, in alcune sequenze lo scostamento sarà superiore al 10% e tali sequenze non verranno collegate. Questi casi possono essere riconosciuti in questo modo: quando FFmatch è fermo si clicchi su menu Misc→RiportaIControlliFalliti.

Nel testo di sotto è possibile vedere, uno dopo l’altro, tutti i controlli falliti. Le righe tipo questa:

```
sift check: 5754 <-> 5756
```

dicono quali frame sono stati controllati tramite sift (in questo caso, il frame 5754 del primo filmato e 5756 del secondo).

Dopo questa riga viene indicato il motivo per cui il controllo è fallito.

Nei casi che ci interessano, nei frame contenuti nelle zone scalate diversamente, il controllo risulta fallito a causa della proporzionalità (viene stampato il messaggio “proportionality check failed”), cioè è stata superata la soglia di tolleranza (che è del 10% se il parametro è 0.9).

In questi casi, abbassando il parametro Proport.thres. il controllo diventa più tollerante, ad esempio con un valore 0.5 verranno accettati scostamenti fino a 50% dai valori medi.

Mettendo quel parametro a 0, il controllo è disabilitato.

Si tenga presente che, se si cambia “Proport.thresh.”, prima di chiedere nuovamente l’analisi, vanno cancellati i controlli falliti (menu

Pulizia→CancellaIControlliFalliti), altrimenti il programma userà i risultati che trova nella cache.

Si tenga presente infine che, con un valore 0,9 del parametro “Proport.thresh.” i valori riportati come “Rapporto orizzontale/verticale” sono molto precisi e attendibili, ma diminuendo quel parametro, i rapporti indicati potrebbero non avere più alcun significato.

Per concludere, dato che le scalature “miste” sono abbastanza rare, conviene lasciare a 0,9 il parametro “Proport.thresh.”, perché è un controllo che può evitare accoppiamenti errati (e anche perché rende attendibili i valori “Rapporto orizzontale/verticale”). Se, però, si nota che abbiamo a che fare con un filmato a scalatura mista, occorre diminuire (o azzerare) il parametro “Proport.thresh.”, cancellare i controlli falliti e far ripartire l’analisi (F5).

Timestamps troppo irregolari (3 novembre 2020)

[Nota: questa sottosezione è un po’ obsoleta perché successivamente ho implementato i buffer di lettura che gestiscono anche quelli che qui avevo chiamato “buchi”. Per il momento la lascio, perché può ancora essere utile per i filmati che hanno effettivamente timestamps troppo irregolari.]

La prima novità della più recente versione di FFmatch è che supporta numerosi formati contenitori (AVI, MKV, MP4, WMV, ecc.) e standard di compressione video (ASP, AVC, HEVC, ecc.), purché soddisfino una condizione: siano a framerate costante e i timestamps dei frames siano effettivamente coerenti con tale framerate.

Ad esempio, in un filmato a 25 fps, la differenza tra i timestamps di due frames consecutivi deve essere pari a 40 ms (1/25); in un filmato a 23.976 fps deve essere pari a 41.7 ms; e così via...

Questo è esattamente ciò che significa “framerate costante” e questo è ciò che avviene di solito. Vedremo che esistono però filmati che dovrebbero essere a framerate costante ma, per ragioni varie, hanno timestamps che violano questa regola.

Innanzitutto, per fissare un po’ le idee, prendiamo un filmato a 25 fps e facciamoci stampare i timestamps dei primi 20 frames:

```
0.04  
0.08  
0.12  
0.16  
0.2
```

0.24
0.28
0.32
0.36
0.4
0.44
0.48
0.52
0.56
0.6
0.64
0.68
0.72
0.76
0.8

Si può notare che la differenza tra due timestamps consecutivi è sempre pari a 0.04, che corrisponde a un framerate costante di 25 fps (1/0.04).

In questo filmato, la regola è rispettata per tutti i frames, dal primo all'ultimo (oltre 130000 frames).

Perché è importante che questa regola venga rispettata?

Perché, se tra due frames consecutivi passa un intervallo di tempo superiore a quanto dovrebbe, viene a saltare la corrispondenza tra la posizione dei campioni audio e i frames video, e ciò produce un'asincronia audio/video. Se le posizioni di frames "saltate" sono molto poche, l'asincronia che viene a crearsi può essere impercettibile. Ad esempio, in un tipico filmato a 25 fps, se viene saltato un solo frame, l'entità dell'asincronia audio/video è di soli 40 ms (a partire dalla posizione del frame saltato) mentre gli scostamenti iniziano a essere percepiti dalla maggior parte delle persone intorno ai 100 ms. Ma se i frames saltati sono diversi, le asincronie si cumulano e il filmato ottenuto, da un certo punto in poi, sarà palesemente con audio fuori sync.

È perciò importante controllare se i timestamps siano o meno coerenti con il framerate.

Il "vecchio" FFmpeg non faceva alcun controllo sui timestamps, dando per scontato che la sequenza dei frames fornita dalle librerie FFMPEG fosse priva di "buchi", cioè che i timestamps fossero appunto regolari, come nell'esempio visto sopra.

Nel vecchio FFmpeg il problema delle asincronie dovute a questo motivo poteva comunque presentarsi molto raramente, perché supportava praticamente soltanto gli AVI (in effetti supportava anche MP4 ma solo a 25 fps) e la maggior parte degli AVI soddisfa la condizione (esistono alcune eccezioni, che vedremo in seguito).

Quando, dopo molti anni, ho deciso di rimettere mano a FFmpeg in modo che leggesse altri formati, mi sono scontrato con i timestamps, perché le librerie FFMPEG non forniscono un modo immediato per accedere a un

frame di posizione arbitraria, ma forniscono un modo, anche se non semplicissimo, per accedere a un frame corrispondente a un determinato timestamp. La soluzione più ovvia, quindi, era di usare una formula per convertire una posizione nel corrispondente timestamp, e questa, sorvolando sui dettagli, consiste nel moltiplicare il numero della posizione per la durata (fissa) di un frame, in altre parole nel rapporto tra numero di posizione e framerate (costante).

Mentre testavo alcune varianti di questa formula, constatavo che funzionava correttamente su molti filmati, ma saltavano fuori alcune eccezioni su cui nessuna variante funzionava. Si noti che se non esiste una formula che converta una posizione nel corretto timestamp, con FFMPEG non è possibile accedere a una posizione arbitraria (dal punto di vista di un player, questo non è un problema perché in un player gli spostamenti vengono fatti su tempi, non su posizioni). In realtà un modo ci sarebbe, ma presuppone una scansione preventiva del filmato, durante la quale si memorizzano i timestamps (indicizzazione); il vettore così definito potrà successivamente essere usato come indice/funzione di conversione tra posizioni e timestamps. I vari formati contenitore, in genere, forniscono già un tale indice, ma le relative funzioni delle librerie FFMPEG non sono affidabili (a volte sono ben definite, come nel caso degli AVI, ma più spesso no).

Sempre per fissare le idee, vediamo i timestamps dei primi 20 frames di un filmato reale in cui mi sono imbattuto durante la prima fase di test. Il filmato è in contenitore MKV, risulta avere un framerate di 29.97 fps, e questi sono i suoi primi timestamps:

```
0
0.033 (0.033)
0.067 (0.034)
0.1 (0.033)
0.133 (0.033)
0.184 (0.051)
0.234 (0.05)
0.284 (0.05)
0.334 (0.05)
0.384 (0.05)
0.434 (0.05)
0.484 (0.05)
0.534 (0.05)
0.584 (0.05)
0.634 (0.05)
0.684 (0.05)
0.734 (0.05)
0.784 (0.05)
0.834 (0.05)
0.884 (0.05)
0.934 (0.05)
```

Tra parentesi ho fatto stampare la differenza del timestamp con quello precedente.

Come si vede, la differenza non è costante (varia in modo incomprensibile tra 0.033 e 0.051) laddove ci saremmo aspettati l'inverso del framerate nominale ($1/29.97 = 0.033$), valore che compare solo nelle prime posizioni.

Se faccio calcolare a un programma il framerate medio per tutta la durata di questo filmato ottengo: 23.976 fps.

Si tratta quindi di un filmato a 23.976 fps, su cui però, per qualche ragione, è memorizzato nel contenitore MKV il valore 29.97 fps (i timestamps sono del tutto incoerenti con tale valore).

A questo punto mi è venuta l'idea di aprire quel file MKV con `mkvtoolnix-gui`, scrivere il valore 24000/1001p nel campo "Durata predefinita/FPS" e rimuxarlo (pulsante "Avvia multiplexing"). Così facendo, ho ottenuto un nuovo MKV che risulta essere a 23.976 fps, che viene riprodotto perfettamente (audio e video sono sincronizzati dall'inizio alla fine) e che ha questi timestamps:

```
0
0.042 (0.042)
0.083 (0.041)
0.125 (0.042)
0.167 (0.042)
0.209 (0.042)
0.25 (0.041)
0.292 (0.042)
0.334 (0.042)
0.375 (0.041)
0.417 (0.042)
0.459 (0.042)
0.5 (0.041)
0.542 (0.042)
0.584 (0.042)
0.626 (0.042)
0.667 (0.041)
0.709 (0.042)
0.751 (0.042)
0.792 (0.041)
0.834 (0.042)
```

Come si nota, i timestamps del nuovo MKV tendono a essere simili a quelli del filmato di partenza (e più si va avanti, più sono simili; da un certo punto in poi sono identici), il che spiega perché i players li riproducono entrambi senza difetti.

Ma il fatto più rilevante è che le differenze tra due timestamps consecutivi è costante, ed è pari all'inverso di 23.976 ($1/23.976 = 0.041708$).

Il caso che abbiamo appena analizzato è quello di un filmato (MKV) che è a framerate costante, ma il framerate memorizzato nel contenitore non è quello vero e soprattutto i timestamps non crescono in modo regolare

(come dovrebbe essere in un filmato a framerate costante), coerentemente con il framerate nominale.

Come abbiamo visto, tuttavia, è bastato rimuxarlo con il muxer ufficiale del progetto Matroska (specificando il valore corretto di framerate) per ottenere un MKV che soddisfa la condizione essenziale di un filmato a framerate costante.

Dato che filmati di questo tipo sono la minoranza (la maggior parte dei file MKV che ho testato ha il framerate nominale corretto e i timestamps sono coerenti con esso) e che con mkvtoolnix-gui è molto facile ottenere una copia “buona” (è un semplice remux, che non richiede alcuna modifica ai bitstream), ho deciso di implementare in FFmatch l’accesso a posizioni arbitrarie mediante l’uso di formula di conversione (da posizione a timestamp), evitando quindi l’indicizzazione.

Quando FFmatch si imbatte in un filmato in cui la crescita dei timestamps non rispetta il framerate, stampa un messaggio (nel testo di sotto) e interrompe la scansione (il messaggio è del tipo *irregular timestamps not supported... Aborted*). A questo punto l’utente può mettere a posto i timestamps con il metodo visto sopra, cioè: apre il filmato avente i timestamps irregolari con mkvtoolnix-gui, digita il valore nel campo “Durata predefinita/FPS” relativo al bitstream video, e salva un nuovo MKV cliccando sul pulsante “Avvia multiplexing”:

The screenshot shows the mkvtoolnix-gui interface. On the left, a table lists tracks with columns for Codec, Tipo, Copia elemento, Lingua, Nome, ID, and Lingua predefinita. The tracks are:

Codec	Tipo	Copia elemento	Lingua	Nome	ID	Lingua predefinita
<input checked="" type="checkbox"/> MPEG-1/2	Video	<input checked="" type="checkbox"/> Sì	en		0	<input type="checkbox"/> No
<input checked="" type="checkbox"/> AC-3	Audio	<input checked="" type="checkbox"/> Sì	fr	2/0	1	<input checked="" type="checkbox"/> Sì
<input type="checkbox"/> AC-3	Audio	<input type="checkbox"/> No	es	2/0	2	<input type="checkbox"/> No
<input checked="" type="checkbox"/> VobSub	Sot...	<input checked="" type="checkbox"/> Sì	en		3	<input checked="" type="checkbox"/> Sì
<input checked="" type="checkbox"/> 11 elem...	Ca...	<input checked="" type="checkbox"/> Sì				

On the right, the "Timestamp e durata predefinita" section contains the following fields:

- Ritardo (in ms):
- Ridimensionato di:
- Durata predefinita/FPS:
- File timestamp:
- Correzione informazione timing flusso

At the bottom, the "File destinazione" section shows:

File destinazione:

Buttons at the bottom:

E se non siamo sicuri di quale sia il corretto framerate?

In realtà, ai fini dell’uso come “copia temporanea” per FFmatch, possiamo inserire un valore qualsiasi nel campo “Durata predefinita/FPS”, magari

togliendo la spunta a tutte le tracce diverse da quella video (in modo che non vengano copiate nel nuovo MKV). Per qualsiasi valore si scelga, si otterrà sempre un filmato con timestamps coerenti con quel valore. Ovviamente valori diversi daranno timestamps diversi, e quindi durate diverse, ma ai fini dell'uso in FFmpeg tutto ciò non ha importanza, perché li vedrebbe sempre come la stessa sequenza di immagini, indipendentemente dal valore scelto.

Quello che abbiamo visto finora è il caso dei filmati che, pur essendo a framerate costante, sono stati muxati in modi anomali. Si risolve con un semplice remux.

Esistono però altre tipologie di filmati con timestamps irregolari.

L'esempio più ovvio è quello di filmati che nascono a framerate variabile, ma quelli non dovrebbero mai essere usati a scopo di muxing.

Un'altra tipologia è data dagli AVI contenenti "buchi".

Il contenitore AVI è piuttosto vecchio e poco sofisticato e in teoria non supporterebbe i framerate variabili. Tuttavia i framerate variabili sono sempre stati simulati negli AVI con tecniche del tipo "drop frame" (ad esempio, tanti anni fa avevo una scheda di acquisizione video che decideva di "droppare" frames, quando l'hardware non era sufficientemente veloce per generarli e salvarli). Questo era ottenuto semplicemente creando una entry di lunghezza 0 nell'indice dell'AVI. Tali entries sono rilevate da AVInaptic e indicate come "Null frames". Esistono programmi che usano i null frames negli AVI per mantenere la sincronia audio/video (ad esempio, mencoder, ma in tale programma i null frames possono essere evitati specificando il filtro video *harddup*; si noti che nella guida a FFmpeg del 2012 avevo messo *harddup* come ultimo anello della catena dei filtri video, proprio per evitare i null frames).

Gli AVI hanno un'altra fonte di problemi, ma più precisamente si tratta di una caratteristica dei bitstream video ASP: in AVInaptic sono indicati come N-VOPs, e questi sono frames che nell'indice dell'AVI risultano avere una lunghezza maggiore di 0 (sebbene molto piccola) ma nel bitstream non codificano un'immagine. Ciò che conta è che le librerie FFmpeg non generano nel flusso di immagini né null frames, né N-VOPs, lasciando in tal modo "buchi" nella sequenza dei timestamps.

Questa tipologia di filmati è piuttosto insidiosa perché, come accennavo all'inizio, può dare luogo ad asincronie audio/video.

Un modo per sapere in anticipo se un filmato AVI appartiene a questa classe è quello di analizzarlo con AVInaptic, farne un rapporto completo e

controllare le voci *Null frames* e *N-VOPs* (quest'ultimo sarà presente solo in caso di bitstream ASP; del resto è solo lì che si possono trovare). I valori corrispondenti a tali voci devono essere entrambi 0. Se uno dei due è maggiore di 0, FFmatch si arresterà durante la scansione segnalando l'irregolarità nei timestamps.

I filmati di questo tipo, purtroppo, vanno ricodificati per eliminare le fonti di asincronie. Di solito, comunque, il problema si presenta con filmati vecchi (o generati da programmi vecchi) e di bassa risoluzione (secondo gli standard odierni), quindi la loro ricodifica è molto veloce.

Compilazione

Questa sezione contiene alcune indicazioni per compilare il programma dal codice sorgente su github.

[fixme - da finire]