

# Guida Completa A Dgindex 1.4.5

Scritta da aytin il 13-01-2006

L'ideale complemento di Avisynth. Il frameserving è servito.

---

## GUIDA COMPLETA A DGINDEX 1.4.5

- scritta da: *...:Aytin:...*

- collaborazione, supporto, revisione: **...:Divxmania Staff:...**

### 1. Cos'è DGindex

DGIndex è un piccolo ma potentissimo tool il cui scopo primario è quello di decodificare streams MPEG-1/2 e creare un indice contenente, per ogni frame, le locazioni e tante altre informazioni, prelevandole dagli header.

DGIndex, insieme ad Avisynth, costituisce il completamento per la realizzazione del frameserving ( **cf. Cap 2. di "Editing non lineare con Avisynth"**). DGIndex costruisce l'indice usato da Avisynth (attraverso un plugin per avisynth, DGDecode, che serve per "aprire" gli streams MPEG-2) per accedere ai frames e processarli.

La coppia (DGIndex, Avisynth), mette a disposizione un ulteriore strato software attraverso il quale un'applicazione può visualizzare e processare streams che altrimenti non potrebbe manipolare (MPEG1/2, VMW, ASF e via dicendo)

Ritornando allo specifico, DGIndex, oltre ad MPEG-1/2, è in grado di decodificare tanti altri tipi di streams, demuxare audio e video, disporre di elementari funzioni di editing video come il filtro sulla luminanza, il crop o la conversione del Colorspace.

La versione di riferimento è l'ultima release stabile, la **1.4.5**. Tutto ciò che seguirà è in parte basato sulla guida di riferimento, alla quale rimando per approfondimenti ulteriori.

[Cap. 2. Esplorando i menù](#)

[Cap. 3. DGDecode: plugin per Avisynth](#)

[Cap. 4. Analisi con DGIndex](#) **2. Esplorando i menù**

La GUI di DGIndex è molto semplice ed intuitiva da usare.

L'ordinamento dei menu riflette una sorta "gerarchia operativa" da seguire durante l'utilizzo.

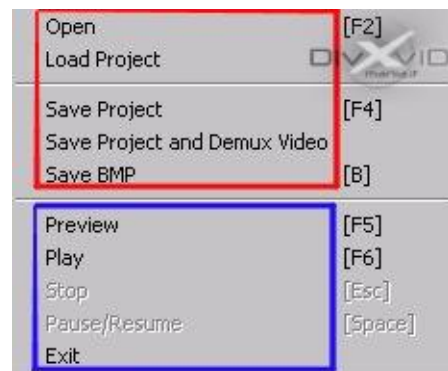
## 2.1. File

Il menù **File** è diviso in 2 parti:

La 1ª relativa al caricamento o al salvataggio del progetto, vale a dire al caricamento dell'indice o alla sua creazione.

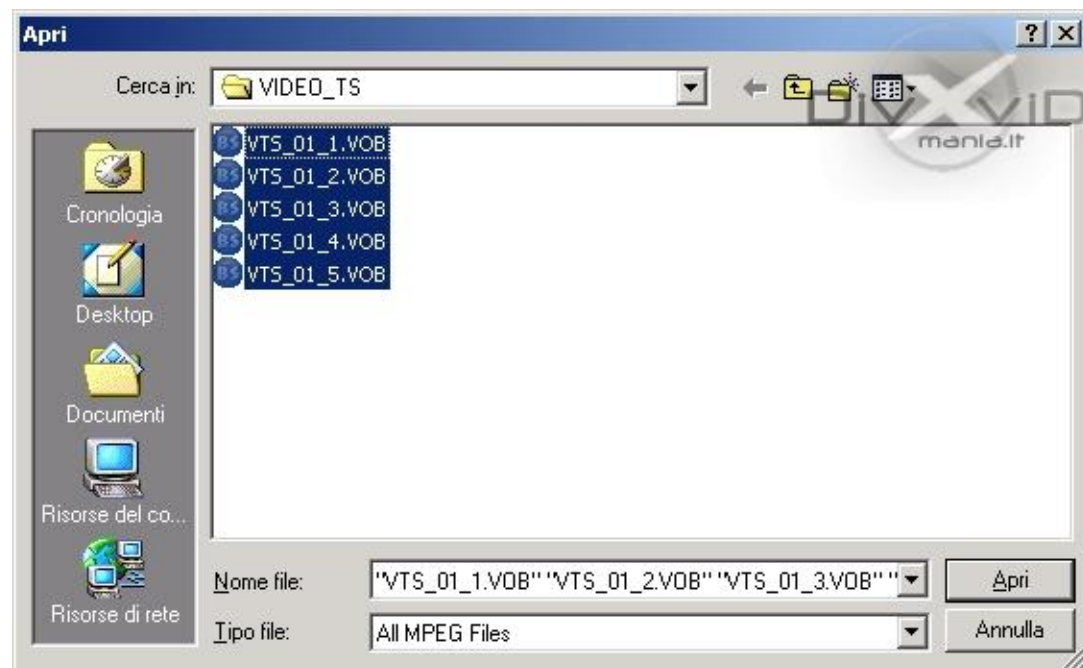
La 2ª parte è relativa alla visualizzazione dell'anteprima.

**Preview** attiva l'anteprima disabilitando gli altri menu, mentre **play, stop, pause/resume** regolano, com'era facile immaginare, il playback.



**Fig. 2.1 - Menù "File"**

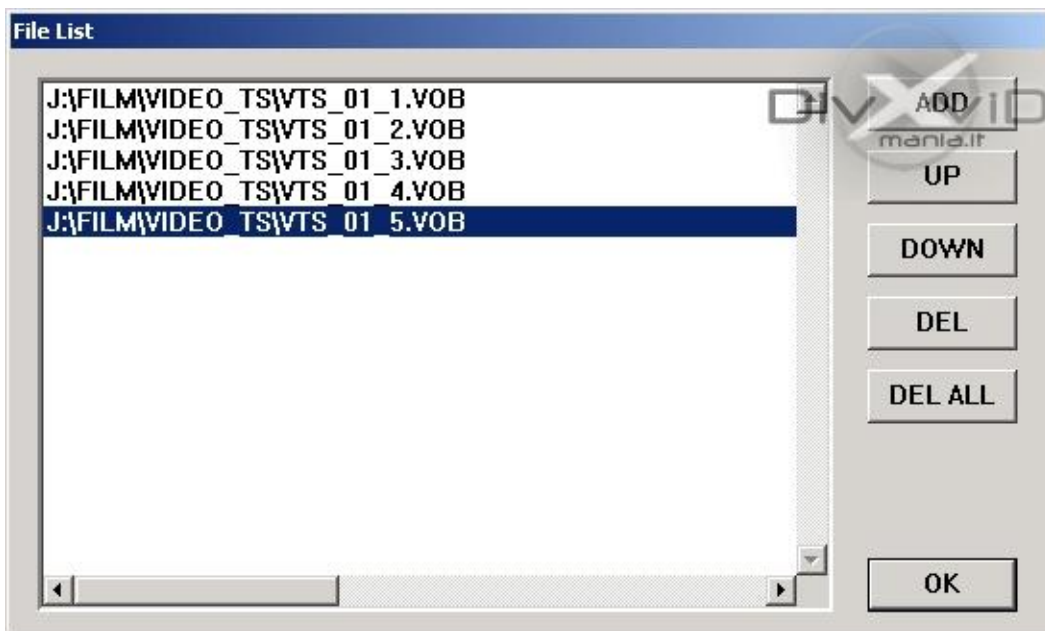
Per la creazione dell'indice (file `.d2v`) occorre indicare, attraverso **Open**, i VOB da includere o i file MPEG-1 da processare.



**Fig. 2.2 - Selezione files**

Load Project carica un file `d2v` salvato in precedenza.

In entrambi i casi il risultato è identico: DGIndex mostrerà la lista di ciò che è stato selezionato.



**Fig. 2.3 - Modifica lista**

Questa lista può essere modificata:

- attraverso i pulsanti **ADD**, **DEL**, **DEL ALL** si possono aggiungere o sottrarre ulteriori files.
- attraverso i pulsanti **UP**, **DOWN** può essere modificato l'ordine di inserimento dei files e di conseguenza, l'ordine in cui i frames verranno processati.

Una volta premuto **OK**, DGIndex è, di fatto, in grado di produrre il nostro file indice, salvo compiere ulteriori settaggi.

La slidebar **(1)** mostrata in figura, serve per selezionare la porzione di video da processare o di cui si vuol visualizzare l'anteprima.

La selezione della porzione video, che il default include completamente, avviene attraverso i pulsanti **(2)** e **(3)** oppure premendo rispettivamente "[" per lo start, oppure "]" per l'end, secondo il consueto meccanismo che conosce bene chi ha usato almeno una volta VirtualDub o similari.

I pulsanti "<" e ">" **(4)**, servono per l'avanzamento frame-to-frame.



**Fig. 2.4 - Slidebar di DGIndex**

- **Save Project** crea il file d2v, oltre che, eventualmente, demuxare l'audio.
- **Save Project and demux Video** credo non abbia bisogno di commenti 😊

Oltre che fare tutto quello che è stato enunciato in precedenza, si demuxa anche il video sotto forma di un file MPEG che sarà:

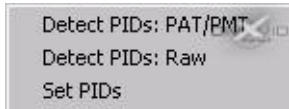
- *.m1v* per gli streams MPEG-1
- *.m2v* per gli streams MPEG-2

Tali files possono essere utili nei programmi di DVD authoring, per esempio.

- **Save BMP**, molto utile in fase di test, permette di salvare il frame corrente come bitmap.

## **2.2. Stream**

DGIndex permette di aprire direttamente gli streams una volta specificati i PIDs audio e video.



**Fig. 2.5 - Menù "Streams"**

Usando **Detect PIDs: PAT/PMT**, viene mostrata una lista di streams audio/video e si scelgono quelli da decodificare selezionando lo stream video e quello audio.

Questa selezione è resa possibile dal fatto che i transport streams, periodicamente, inviano delle tabelle di dati, le tabelle PAT/PMT per l'appunto, contenenti i riferimenti (PIDs) per gli streams audio e video.

A volte però queste tabelle di dati non sono disponibili ed allora è necessario ricorrere ai dati grezzi, senza riferimenti. I cosiddetti raws data.

**Detect PIDs: Raw**, individua gli streams così come si trovano, senza nessun riferimento di sorta, e dal momento che non si può capire quali siano gli stream audio e video che vanno insieme, si deve procedere per tentativi.

**Set PIDs**, infine, permette di settare manualmente i pids. DGIndex, in questo caso non effettuerà nessun tentativo di "detect"

## **2.3. Video**

Anche il menù **Video**, si può immaginare diviso in 2 parti.



**Fig. 2.6 - Menù "Video"**

La 1<sub>a</sub> parte (selezione rossa), comprende due voci che sono forse fra le più importanti da settare su DGIndex.

La 2<sub>a</sub> parte (selezione blu) è quella che permette un certo filtraggio sul video.

Analizziamo prima la 2<sub>a</sub> parte.

Nel caso in cui il frameserve avvenga tramite DGVfapi (invece che con DGdecode, come vedremo in seguito, per il quale questo settaggio è ininfluente) il video è sempre convertito in RGB.

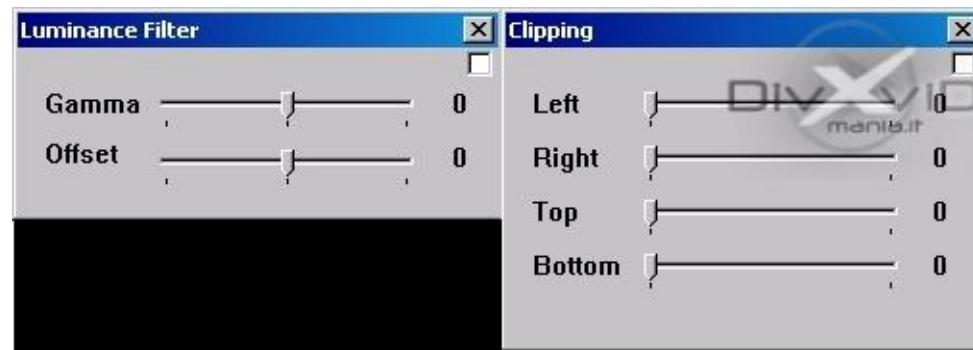
**YUV -> RGB** effettua una conversione del colorspace su due scale differenti:

- YUV [16, 235(Y)/240(UV)] => RGB [0, 255] (**PC scale**)
- YUV [16, 235(Y)/240(UV)] => RGB [16, 255] (**TV scale**)

Al 98% si utilizza la 1<sub>a</sub>, **PC scale**, tranne che in pochissimi casi consigliati dalla documentazione

completissima inclusa in DGIndex.

**Luminance Filter** e **Clipping** permettono di regolare la luminosità, un pò come avviene con alcuni filtri per avisynth, e di croppare l'immagine.



**Fig. 2.7 - Luma e Crop**

Le slidebar regolano l'intensità delle impostazioni e la selezione della checkbox in alto a destra consente di visualizzare direttamente sull'anteprima le conseguenze dei settaggi appena fatti.



**Il consiglio è di non fare uso di questi filtri.**

**Visto che l'idea di fondo è quella di fare il frameserving attraverso DGDecode, quindi con Avisynth, si avrà l'opportunità di intervenire meglio e con più precisione in seguito (cfr. Cap. 4 di "Editing non lineare con Avisynth")**

Veniamo al punto cruciale, i settaggi fra i più importanti di DGIndex.

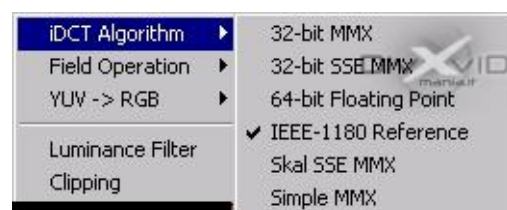
Semplicissimi da effettuare ma un errore in questo punto potrebbe compromettere l'intera conversione.

- **iDCT Algorithm**
- **Field Operation**

### **iDCT ALGORITHM**

Nella versione 1.4.5 di DGIndex, ci sono ben 7 algoritmi a disposizione. Gli algoritmi non supportati dal processore usato vengono automaticamente esclusi dal menu (nella figura in basso, manca l'algoritmo **32-bit SSE2 MMX** perchè non supportato dal mio processore).

La scelta dell'algoritmo iDCT può essere fatta in base a criteri di velocità o di qualità:



**Fig. 2.8 - Scelta dell'algoritmo iDCT**

La natura dei processori attuali permette di dare maggiore rilievo a quest'ultima, a differenza di quanto avveniva in passato quando si doveva cercare un buon compromesso.

Nella documentazione allegata c'è un'utilissima classifica degli algoritmi iDCT che può essere il riferimento

per chi cerca di capire quale sia l'algoritmo più adatto.

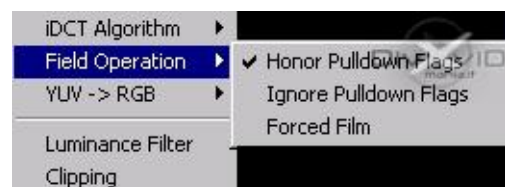
- Qualità: **IEEE-1180 Reference** > **64-bit Floating Point** > tutti gli altri iDCT
- Velocità: **32-bit SSE2 MMX** è il più veloce. **IEEE-1180 Reference** è il più lento. Gli altri sono più o meno allo stesso livello fra loro.

Per un approfondimento sull'iDCT rimando a [questo topic](#) di Divxmania, scritto da CereS.

Credo sia stata una delle prime cose ad aver letto quando cercavo info su cosa fosse l'iDCT e non ho mai trovato niente di altrettanto chiaro e sintetico.

## **FIELD OPERATION**

Tre semplici voci, la cui selezione dev'essere molto accurata:



**Fig. 2.9 - Field Operation**

**Field Operation**, permette di decidere come gestire il flag RFF (REPEAT\_FIRST\_FIELD) che, nello stream MPEG-2, serve per replicare il frame che contiene tale flag al fine di realizzare il **3:2 pulldown**.

Il **3:2 pulldown** è utilizzato per permettere ai sistemi NTSC (29,97 fps) la visualizzazione di video a 23,976 fps.

Non mi dilungherò oltre sulla spiegazione di cosa sia il **3:2 pulldown**.

Una spiegazione più che esaustiva dell'argomento si trova nei paragrafi **6.1**, **6.2** e **6.3** di "[Editing non lineare con Avisynth](#)", ai quali vi rimando per una comprensione totale di tutto quello che seguirà.

**Honor Pulldown flags** e **Ignore Pulldown Flags** hanno comportamenti antitetici.

Se **Honor Pulldown flags** è settato, il file d2v prodotto conterrà i riferimenti anche agli eventuali frame ripetuti, relativamente ai flags RFF presenti, altrimenti (settando **Ignore Pulldown Flags**) l'indice verrà costruito in base ai frame così come sono, senza tenere conto dei flags RFF

Nel 1° caso:

- per sorgenti MPEG-2 provenienti da area NTSC, l'indice verrà generato in modo da essere utile per la creazione di un file NTSC perchè rifletterà la presenza dei frames ripetuti.
- per sorgenti MPEG-2 provenienti da area PAL e streams MPEG-1, non avendo flag RFF, l'applicazione di **Honor Pulldown flags** non presenta alcuna conseguenza.

Nel 2° caso, non tenendo conto dei flags RFF, l'indice si riferirà invece ai frames così come si presentano, senza ripetizioni di sorta.

Che bisogna fare?

Se si vuole rispettare la natura della sorgente, occorre **sempre** lasciare **Honor Pulldown flags** settato.

Per quel che ci riguarda, è quello che bisogna fare anche per i DVD di area PAL.

Ignorare i flags, settando **Ignore Pulldown flags**, può essere dannoso anche per gli utenti di area PAL. Infatti, se per sfortuna ci si trova di fronte ad un pulldown fatto male, la conseguenza è un'asincronia audio-video (sempre che il demultiplex della traccia audio si faccia tramite DGIndex, chiaramente)

**Ignore Pulldown flags** è utile, semmai, in fase di analisi del flusso MPEG-2 originario.

Infine veniamo alla terza e ultima opzione: **Forced Film**

Questa opzione permette di abbassare il framerate agli originali 23,976 fps, nel caso di **FILM** che abbiano subito un 3:2 pulldown.

In pratica non fa che operare un **inverse telecine** (*iVTC*)

Verranno ignorati i flags RFF e inseriti (o rimossi) alcuni frames, al fine di mantenere costante il framerate in uscita, oltre a che a mantenere la sincronia fra audio e video.

Usare questo opzione per video che non siano **FILM** può causare jerk-effect (movimenti a scatti) e/o asincronie audio-video.

Quando usare **Forced Film**?

Diventa molto importante il valore del campo *Video Type* nell'anteprima .

- Se il campo Video Type mostra un valore pari a "**FILM 95%**" o una percentuale superiore, è il caso di usare **Forced Film**.

L'ideale sarebbe avere un 100% per avere la certezza di evitare possibili artefatti. In ogni caso la percentuale del 95% rappresenta una soglia accettabile.

- Se la percentuale è più bassa ma il valore "**FILM**" è tutto sommato rilevante, si può provare ad utilizzare **Forced Film** e se ci sono eventuali frame interlacciati, si interviene con un deinterlace adattivo come *FieldDeinterlace*.

Se il risultato non è soddisfacente, occorre cambiare strategia. Rispettare i flags RFF innanzitutto, settando **Honor Pulldown flags**, e intervenire puntualmente con gli approcci previsti per i **video ibridi**.

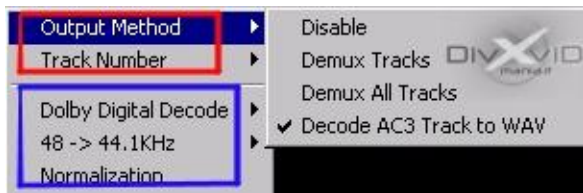
- Se la percentuale è troppo bassa oppure il video è di tipo NTSC, conviene settare **Honor Pulldown flags**, e intervenire, se serve, con dei filtri deinterlacciatori oppure facendo un *iVTC* di altro tipo. Cosa plausibile quando il pattern usato non è il 3:2 pulldown ma un pattern differente a cui si deve risalire analizzando il file *.d2v*(vedere il Cap. 4)

Ricordo sempre il rimando ai paragrafi **6.1**, **6.2** e **6.3** di "[Editing non lineare con Avisynth](#)" per tutto quei concetti che non vengono approfonditi in questo paragrafo e che sembrano essere dati per scontati.

## **2.4. Audio**

Il settaggio dei parametri visti finora potrebbe essere già sufficiente per avere il nostro progettino *.d2v*. Se abbiamo bisogno di demultiplexare anche l'audio, allora tornano utili anche le funzionalità offerte da questo menù.





**Fig. 2.10 - Menù "Audio"**

Il menù **Audio** ha una struttura molto simile a quella del menù **Video**:

- una parte deputata al demultiplex delle tracce audio
- un'altra, ad un minimo di editing delle tracce audio estratte.

**Output Method** e **Track Number** specificano cosa estrarre con esattezza.

A seconda della selezione, si abiliteranno o meno le altre parti del menù **Audio**.

- *Demux All tracks* estrae tutte le tracce così come sono. In questo caso, sia *Track Number* che le altre voci del menu **Audio** saranno disabilitate.
- *Demux tracks* permette di scegliere quali tracce estrarre attraverso *Track Number*, che stavolta risulterà abilitato (es. potremo scegliere la 1, la 2 e la 5). Sarà ancora disabilitata la 2<sup>a</sup> parte del menù (quella in blu).
- *Decode AC3 Track to WAV* abilita la 2<sup>a</sup> parte del menù che permette di effettuare delle normalizzazioni o delle conversioni del sample rate oltre che a convertire direttamente le tracce AC3 estratte in WAV.

**Se lo stream è un MPEG-2, le tracce demultiplexate da *Demux Tracks* e *Demux All Tracks* saranno dei files AC3.**

**Se lo stream è un MPEG-1, le tracce demultiplexate da *Demux Tracks* e *Demux All Tracks* saranno dei files .mpa. La selezione di *Demux AC3 to WAV* non produrrà ovviamente nessun file in uscita.**

Nella conversione della traccia AC3 in WAV entrano in gioco un bel pò di altri fattori.

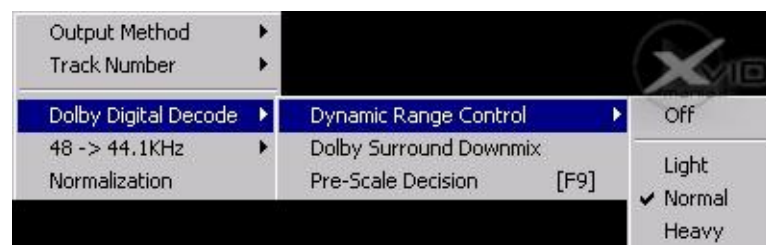
Il **Dolby Digital Decode**, nell'economia della conversione della traccia audio, è importante per la gestione del dolby digital su una traccia in uscita a due soli canali.

## **DYNAMIC RANGE CONTROL**

Nei decoder, il **DRC** è un insieme di metadati. Viene inviato insieme al segnale audio per settare il range dinamico entro cui il segnale audio lavora. Diminuendo il **DRC**, si restringe tale range.

Questo serve per garantire un buon ascolto in ambienti ricchi o meno di rumore di fondo.

Noi faremo la stessa cosa. A seconda dell'ambiente di riferimento, estrarremo la traccia audio con un **DRC** disabilitato oppure che restringa, via via, il range dinamico.



**Fig. 2.11 - Dynamic Range Control**



Si tenga presente che più il range dinamico è ristretto e più il suono risulterà appiattito, uniforme.

## DOLBY SURROUND DOWNMIX

Lo stream AC3 può contenere fino a 6 canali. Il WAV solo due canali stereo, L/R.

Qualcosa si dovrà perdere, nella fattispecie il *surround L/R*, che attraverso il downmixing finirà nei canali stereo L/R.

Il downmix con DGIndex si può ottenere in due modi: attivando o meno la voce corrispondente.

- nel 1° caso verrà effettuato un downmix surround-compatibile.
- nel 2° caso verrà effettuato un *simple-downmix* stereo-compatibile.

## PRE-SCALE DECISION

E' una normalizzazione più raffinata.

- Per prima cosa si setta la normalizzazione senza però selezionare la checkbox.
- Poi si seleziona *Pre-Scale Decision*.

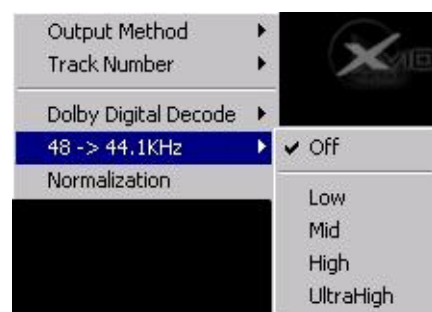
Verrà compiuta una prima scansione, che servirà per determinare un fattore di amplificazione ( *gain factor*) della traccia audio.

L'audio verrà amplificato e **su tale base** verrà calcolato il peak, ossia il massimo livello audio ottenibile tenendo conto della percentuale di normalizzazione.

- Il salvataggio del progetto concluderà tutte le operazioni come al solito

E' una normalizzazione su un audio amplificato che trova le sue ragioni nella riduzione degli errori di quantizzazione per il decoding del dolby digital.

Oltre al **Dolby Digital Decode**, possiamo modificare il sample-rate da 48 KHz a 44.1 KHz



**Fig. 2.12 - Modifica Sample-Rate**

La precisione della conversione si paga con un maggior tempo di codifica.

Infine abbiamo **Normalization**

Tipicamente, serve per ridurre i suoni più forti nella misura della percentuale fissata nella finestra di configurazione.



## Fig. 2.13 - Normalizzazione

Il salvataggio del progetto avverrà in 2 passi.

Nel 1° si estrae l'audio così com'è. Nel 2° si normalizza.

Si tenga conto che ogni tipo di normalizzazione dell'audio toglie spazialità al suono, dovendone livellare i picchi.

A questo punto, settando audio e video (o solo *Video* e disabilitando l'*Audio*, come succede spesso) siamo in grado di creare il nostro progetto *.d2v* premendo F4 o selezionando *Save Project* dal menù **File**.

DGIndex inizierà a lavorare e a noi non rimarrà che continuare con Avisynth, una volta finito.



Fig. 2.15 - Creazione file d2v

## 2.5. Options

C'è poco da dire su questo menù.

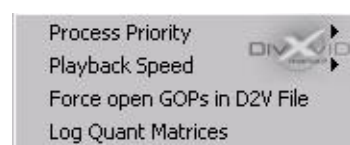


Fig. 2.16 - Menù "Options"

**Process Priority:** assegna la priorità di DGIndex al processore. A meno che non il PC non sia impegnato con qualcos'altro, il che giustificerebbe una diminuzione delle priorità, *normal* (il default) va più che

bene.

**Playback Speed:** `[/i][/b]` regola la velocità di riproduzione durante l'anteprima. Da un minimo, rappresentato dall'avanzamento frame-to-frame, al massimo consentito dal processore.

**Force open GOPs in D2V File:** capita che gli streams non siano codificati correttamente e che alcuni GOPs appaiano chiusi quando in realtà sono aperti. La conseguenza è la comparsa di artefatti sotto forma di blocchetti quando si accede in maniera casuale al video durante la riproduzione in frameserving (script avs per es.).

Per ovviare a questo inconveniente basta selezionare questa voce.

Poichè tale selezione influisce sulle performance, sarebbe meglio farla quando c'è effettiva necessità.

- creare normalmente un progetto *d2v*
- verificare nel frameserving la comparsa di questi artefatti
- se si presentano, creare un nuovo progetto *d2v* attivando l'apertura forzata dei gop.

**Log Quant Matrices:** ha uno scopo puramente "didattico". Serve per dare evidenza alle matrici di quantizzazione che sono state usate nel processo di codifica dello stream MPEG-2. Possono essere diverse, perchè una può lavorare meglio di un'altra, a secondo della natura del video. Il file viene salvato in formato *.txt*.

## 2.6. Tools

Contiene un'unica voce **Parse D2V**, sul cui utilizzo ritornerò più diffusamente in seguito, che produce un file di testo sulla base del file *d2v*.

Tale file estrapola dal file *.d2v* alcune informazioni molto utili in fase di analisi dello stream MPEG-2

[Cap. 1. Cos'è DGIndex](#)

[Cap. 3. DGDecode: plugin per Avisynth](#)

[Cap. 4. Analisi con DGIndex](#) **3. DGDecode: plugin per Avisynth**

Per il frameserving attraverso Avisynth, DGIndex mette a disposizione una dll che andrà importata nello script avs tramite loadPlugin

### CODE

```
loadPlugin (c:\program files\DGMPGDec\DGDecode.dll")
```

La dll comprende 4 filtri:

- *MPEG2Source()*
- *LumaYV12()*
- *BlindPP()*
- *Deblock()*

### **3.1. MPEG2Source()**

Il core del frame-serving offerto da DGDecode, è il filtro che permette di caricare gli streams MPEG-1/2. MPEG2Source ha ben 13 parametri ma nel 99% dei casi ne verranno usati solo 2 (i primi 2)

- **d2v:** *path\file.d2v*  
è il riferimento al file prodotto da DGIndex
- **idct:** *0..7* **[default: 0]**  
specifica l'iDCT da usare relativamente al processore in uso. Il valore 0 usa l'iDCT indicato nel file *.d2v* (vedi par. 4.2)
- **cpu:** *0..6* **[default: 0]**  
Effettua un deblocking su vari livelli (luma, chroma, horizontal, vertical)
- **ipp:** *true/false* **[default: auto]**  
forza il post-processing rispettivamente su field-based (*interlaced*) o frame-based (*progressive*).
- **moderate\_h, moderate\_v:** *0..255* **[default: moderate\_h=20, moderate\_v=40]**  
Fissa la sensibilità per l'individuazione dei blocchi. Valori piccoli sono già molto efficaci. Da usare con cautela.
- **cpu2:** **[default: ""]**  
Con una stringa di 6 caratteri speciali ("x", "o") si abilita o meno un deringing o un deblocking su vari livelli
  - 1<sub>a</sub> posizione: deblocking sul luma orizzontale
  - 2<sub>a</sub> posizione: deblocking sul luma verticale
  - 3<sub>a</sub> posizione: deblocking sul chroma orizzontale
  - 4<sub>a</sub> posizione: deblocking sul chroma verticale
  - 5<sub>a</sub> posizione: deringing sul luma
  - 6<sub>a</sub> posizione: deringing sul chromail carattere "x" abilita, il carattere "o" disabilita.  
Ad es. un solo deblocking sul chroma corrisponde a: "ooxxoo"
- **upConv:** *0..2* **[default: 0]**  
Cambia il formato colore in YUY2 - 4:2:2 (*upConv=1*) o in RGB24 (*upConv=2[i]*)
- **icc:** *[i]true/false* **[default: auto]**  
esattamente come **ipp**, solo che dev'essere attivo l'upSampling mode (*upConv = 1* o *upConv = 2*)
- **i420:** *true/false* **[default: false]**  
Cambia il ColorSpace. Se l'upSampling mode è attivo, questo settaggio non avrà effetto
- **info:** *0..3* **[default: 0]**  
Sono informazioni di debug.
  - Nessuna informazione di debug
  - Mostra le informazioni di debug a video
  - Mostra le informazioni di debug via *OutputDebugString()*
  - Mostra gli hints a video
- **showQ:** *true/false* **[default: false]**  
Altre informazioni di debug. Mostra i quantizer dei macroblocchi.
- **fastMC:** *true/false* **[default: false]**  
Efficiente motion compensation che aumenta un pò le prestazioni a fronte di un certo degrado qualitativo.  
Utilizzato essenzialmente in fase di test e destinato a scomparire nelle future versioni.

- **offset:** -255..+255 **[default: 0]**

valori negativi scuriscono il video, valori positivi lo rendono più chiaro

- **gain:** 0..2 **[default: 1]**

Amplifica l'effetto dell'offset, scurendo il video per valori <1, rendendolo più chiaro per valori > 1

Per maggiori approfondimenti cfr. **Cap. 4. di "Editing non lineare con Avisynth"**

### **3.3. BlindPP()**

Fornisce una funzionalità di deblocking, esattamente come si poteva fare con MPEG2Source.

- **quant:** 0..31 **[default: 2]**

Specifica l'intensità del deblocking

- **cpu:** 0..6 **[default: 6]**

Identico a MPEG2Source ma con default differente.

- **iPP:** true/false **[default: false]**

Identico a MPEG2Source. A differenza dell'analogo parametro su MPEG2Source, non ha valori "auto"

- **moderate\_h, moderate\_v, cpu2:** 0..255 **[default: moderate\_h=20, moderate\_v=40]**

Identici a MPEG2Source.

### **3.4. Deblock()**

Siccome non ci basta mai, ancora un filtro di deblocking.

- **quant:** 0..51 **[default: 25]**

Specifica l'intensità del deblocking

- **aOffset:** 0..51 **[default: 0]**

Setta l'efficacia dell'individuazione dei blocchi. Valori più alti renderanno il deblocking via via più intenso. I bordi saranno sempre più spianati.

- **bOffset:** 0..51 **[default: 0]**

E' un modificatore per l'intensità del deblocking e per l'individuazione dei blocchi. Valori più alti, rendono il filtro più intenso.

- **mmx:** true/false **[default: true]**

Abilita o meno le ottimizzazioni MMX

- **isse:** true/false **[default: true]**

Abilita o meno le ottimizzazioni SSE

Ovviamente la proprietà più importante di questo insieme di filtri è la capacità di acquisire streams MPEG-1/2.

Ad arricchire questa peculiarità, vengono fornite ulteriori funzionalità di deringing, deblocking e luma, peraltro disponibili su altre decine di filtri Avisynth.

[Cap. 1. Cos'è DGIndex](#)

[Cap. 2. Esplorando i menù](#)

[Cap. 4. Analisi con DGIndex](#) **4. Analisi con DGIndex**

DGIndex oltre che a fornire 'sto benedetto file .d2v è molto utile in fase di analisi dello stream, in particolare per quello MPEG-2

Tale analisi si può effettuare in 3 modi differenti che non sono mutuamente esclusivi. Anzi, spesso finiscono per essere complementari.

Tali sistemi sono:

- visualizzazione dell'anteprima:  
Fornisce un gran numero di informazioni, sulla natura del video, frame-rate, colorSpace, frame-type, ecc.
- esame del file d2v:  
C'è praticamente ogni informazione dello stream catturata negli header
- esame dell'output prodotto da *parseD2V*:  
Rappresenta un "sunto" di ciò che può essere già reperito sul file .d2v. Non per niente, il tool lavora proprio su questo file.

#### **4.1. Utilizzo dell'anteprima**

La visualizzazione dell'anteprima può avvenire

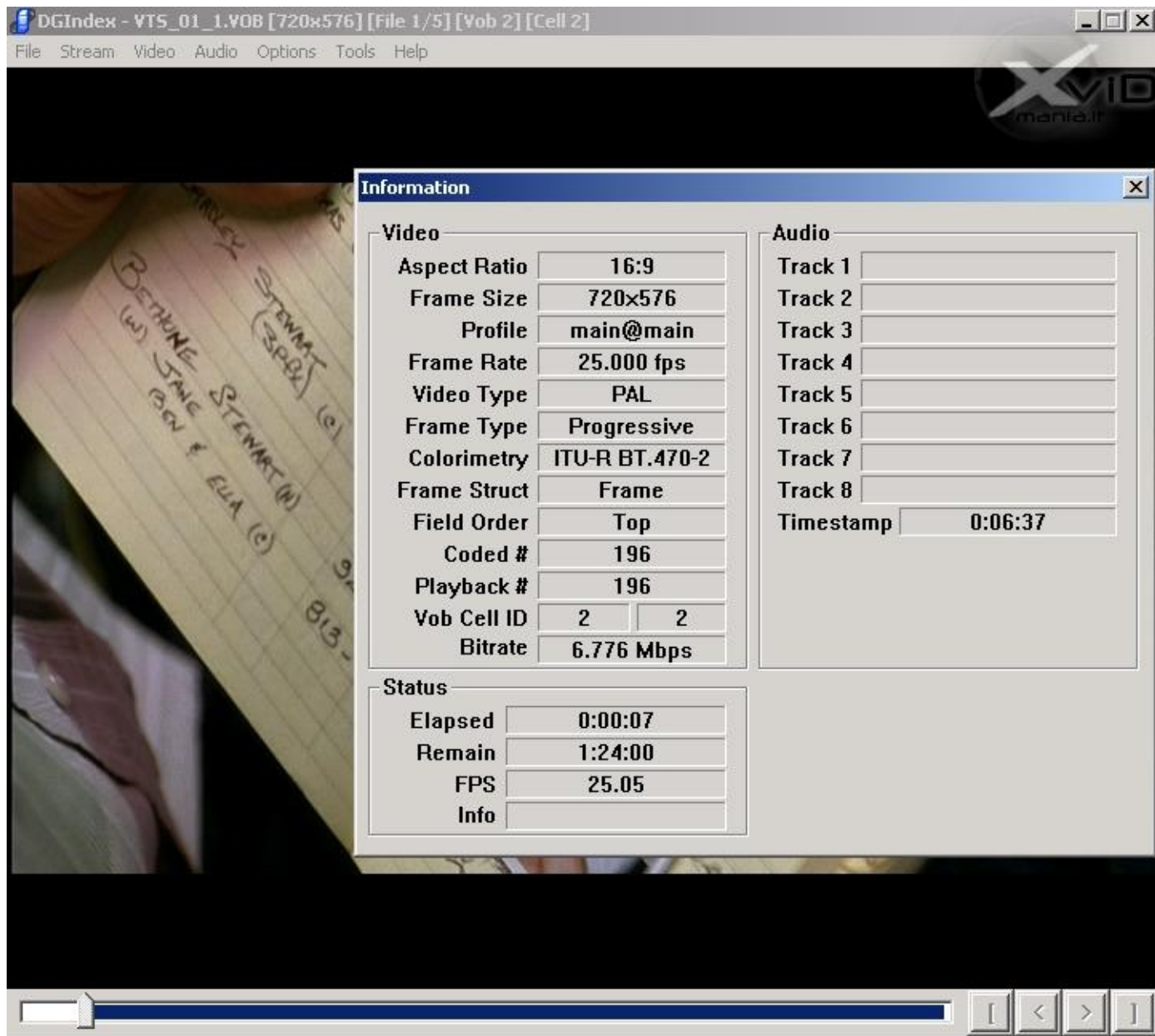
- dopo aver caricato i VOB,
- oppure dopo aver caricato i files MPEG-1,
- o ancora, dopo aver caricato un progetto .d2v tramite **Load Project**.

Il playback avviene selezionando **Preview** nel menu **File** o premendo direttamente F5 (vedi par. 2.1.)

Per utilizzare l'anteprima per l'analisi, è meglio posizionarsi un pò dopo l'inizio del film, com'è mostrato in figura (ved. "selezione porzione video")

Questo perchè i logo dei films possono essere encodeate in maniera differente rispetto al film.

e di conseguenza informazioni come **colorimetry**, **video-type** e a volte del **frame-type**, possono essere molto differenti.



**Fig. 4.1 - Visualizzazione anteprima**

Ora lasciamo che l'anteprima lavori un paio di minuti.

Noi non dovremo far altro che osservare il riquadro **Information** e prendere nota di alcuni valori significativi.

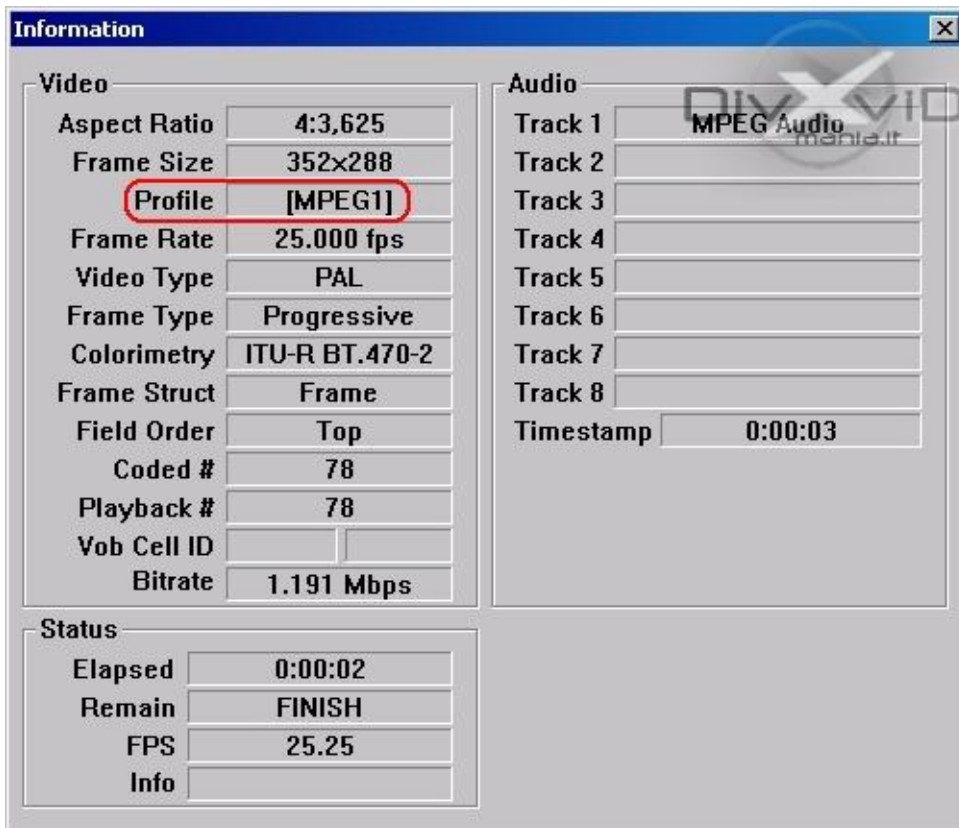
Nella maggior parte dei casi, 10 secondi saranno più che sufficienti 😊

L'anteprima è composta da 3 sezioni: **Video**, **Audio** e **Status**. Ora passeremo al setaccio tutti i campi.

## **VIDEO**

- **Aspect Ratio:** Mostra semplicemente l'AR ad es. 4:3, 16:9 o 2.21:1. A seconda che si tratti di MPEG-1 o MPEG-2 possiamo avere insieme di valori differenti per i quali vi rimando alla documentazione ufficiale di DGIndex irraggiungibile dal menù **Help**.
- **Frame Size:** E' la risoluzione. Nei DVD commerciali contenenti un video MPEG-2 il valore tipico è 720 x 576 (o 720 x 480).
- **Profile:** Corrisponde al modi encodare i dati del video. Se abbiamo un video MPEG, il valore di *Profile* sarà **MPEG1** (vedi fig.2 e nota anche come può cambiare la risoluzione).





**Fig. 4.2 - Anteprima MPEG-1**

- **Frame Rate:** E' il framerate del video.
- **Video Type:** Se non ci sono i flags del pulldown e il framerate corrisponde a 25, *Video Type* assumerà il valore **PAL**, altrimenti **NTSC**.  
Se sono presenti i flags del pulldown il campo mostrerà come valore **FILM** in percentuale oppure **NTSC**, sempre in percentuale.



### Fig. 4.3 - Video NTSC

Alla fine del processo verrà mostrato il valore più grande dei due e sulla base di questo valore andranno fatte le considerazioni riportate nel caso "Field Operation" del par. 2.3

- **Frame Type:** assume come valori **Interlaced** o **Progressive**.

Attenzione. Il valore di questo campo **non stabilisce** che un video sia *interlaced* o

**progressive** [/i] ma è la lettura dell'ennesimo flag (**PROGRESSIVE\_FRAME**) del frame che indica [b] in quale modo il medesimo sia stato encodato: come *progressive* o come *interlaced*

.  
Non dice nulla sul reale contenuto del frame ma semplicemente riporta il valore del flag relativo.

Se un video è **veramente** *interlaced*, il valore di **Frame Type** sarà *interlaced* come ci aspettiamo.

Il viceversa non vale.

E' molto comune trovare dei video che sono *Progressive* ma che l'anteprima di DGIndex fa risultare come *Interlaced*.

Per trovare traccia di interlacciamenti occorre visualizzare un anteprima del video, procedere frame per frame (nelle scene in movimento soprattutto) e verificare che compaia la tipica scalettatura dei video interlacciati. Per maggior completezza sul trattamento di video interlacciati, vedi anche **par. 6.1 e par. 6.6. di "Editing non lineare con Avisynth"**.

- **Colorimetry:** Mostra lo schema dei coefficienti di colori usato. La conoscenza di tale schema è molto utile in fase di codifica perchè è possibile rettificare il modo di rappresentare i colori, onde evitare il leggero sbiadimento tipico dei codec MPEG-4. Al solito, si rimanda al **par. 7.1 di "Editing non lineare con Avisynth"** per una trattazione completa dell'argomento.



**Prestare attenzione alla versione di DGIndex in uso.**

**Le precedenti versioni di DGIndex hanno avuto bachi ricorrenti sia riguardo l'individuazione del corretto Colorimetry che sul demux delle tracce audio.**

**La versione 1.4.5, che è quella a cui si fa riferimento in questa trattazione, non presenta questo tipo di anomalie.**

- **Frame Struct:** Un frame può essere strutturato per **frame** o per **field** quando un frame contiene l'immagine intera o una parte di essa (i campi)
- **Field Order:** Indica l'ordine di visualizzazione dei campi. Può valere **top** o **bottom**. E' pertinente quando **Frame Struct** vale *field* perchè indica in quale ordine visualizzare i campi. Se il frame contiene l'immagine intera, **Field Order** perde di significato.
- **Coded #:** E' il numero di immagini **decodificate** da DGIndex durante l'anteprima o il salvataggio del progetto.
- **Playback #:** E' il numero di immagini **mostrate** da DGIndex durante l'anteprima o il salvataggio del progetto.  
Questo valore può essere maggiore di **Coded #** se vi è un 3:2 pulldown. In questo caso infatti alcuni frame vengono decodificati una sola volta ma mostrati più di una volta per la realizzazione del pulldown.
- **Vob Cell ID:** Valido solo per gli MPEG-2 su VOB, questo valore mostra la posizione del frame corrente all'interno della matrice (VOB, CELL)
- **Bitrate:** Calcola il bitrate medio ogni 64 frame

- **Track i:** Mostra il tipo di audio e il suo bitrate
- **Timestamp:** Mostra il timestamp per la posizione corrente dello stream audio

## STATUS

- **Elapsed:** Mostra il tempo trascorso dall'inizio del salvataggio del progetto o dall'avvio dell'anteprima.
- **Remain:** Mostra il tempo rimanente al completamento del salvataggio del progetto o alla conclusione dell'anteprima.
- **FPS:** E' il framerate medio relativamente però alla velocità con cui DGIndex processa i frame. Se la velocità di playback per es. viene variata attraverso l'apposito menù, tale valore rifletterà questa variazione
- **Info:** Info di Debug o eventuali informazioni di errore

I valori di cui si deve tener conto, in definitiva, sono quelli relativi al VIDEO.

Vediamo quali valori considerare e come:

- Il **Profile** fornisce le prime informazioni relative alla natura dello stream: MPEG-1/2.
- La coppia (**Frame Rate - Video Type**) ci suggerisce o meno la presenza di un *3:2 pulldown*.
- La coppia (**Frame Type - Frame Struct**) ci suggerisce invece della possibile presenza di video *interlaced* o *progressive*.
- Il **Colorimetry** che indica se sia necessario o meno rettificare la gestione dello schema dei colori in fase di codifica.

Non a casa si parla di "suggerimenti". L'anteprima fornisce delle informazioni sommarie, ricavate perlopiù dai flag associati ai frame e che a volte possono essere fuorvianti.

E' sicuramente un primo passo (che molto spesso è più che sufficiente. Ciò che viene mostrato nel 98% dei DVD commerciali con l'eccezione del **Colorimetry** che può variare, è mostrato in Fig. 4.1), ma nei casi più micragnosi (NTSC e derivati) è obbligatorio completare con la visualizzazione del video frame-to-frame e con l'analisi del file *.d2v* (nel capitolo successivo)

## 4.2. Dettaglio del file .d2v

Quando si salva il progetto d2v ciò che viene fuori è qualcosa del genere:

Esempio:

**CODE**

```
DGIndexProjectFile11
5
J:\FILM\VIDEO_TS\VTS_01_1.VOB
J:\FILM\VIDEO_TS\VTS_01_2.VOB
J:\FILM\VIDEO_TS\VTS_01_3.VOB
J:\FILM\VIDEO_TS\VTS_01_4.VOB
J:\FILM\VIDEO_TS\VTS_01_5.VOB
```

```
Stream_Type=1
MPEG_Type=2
iDCT_Algorithm=5
YUVRGB_Scale=1
Luminance_Filter=0,0
Clipping=0,0,0,0
Aspect_Ratio=16:9
Picture_Size=720x576
Field_Operation=0
Frame_Rate=25000
Location=0,0,4,2F0EE
```

```
d00 1 0 2048 1 1 92 f2 e2 f2 f2 e2 f2 f2 e2 f2 f2 e2
900 1 0 184320 1 1 72 72 d2 f2 f2 e2 f2 f2 e2 f2 f2 e2
900 1 0 372736 1 1 72 72 d2 f2 f2 e2 f2 f2 e2 f2 f2 e2
900 1 0 593920 1 1 72 72 d2 f2 f2 e2 f2 f2 e2 f2 f2 e2
900 1 0 819200 1 1 72 72 d2 f2 f2 e2 f2 f2 e2 f2 f2 e2
900 1 0 1101824 1 1 72 72 d2 f2 f2 e2 f2 f2 e2 f2 f2 e2
900 1 0 1290240 1 1 72 72 d2 f2 f2 e2 f2 f2 e2 f2 f2 e2
900 1 0 1515520 1 1 72 72 d2 f2 f2 e2 f2 f2 e2 f2 f2 e2
900 1 0 1730560 1 1 72 72 d2 f2 f2 e2 f2 f2 e2 f2 f2 e2
...
```

La prima parte è piuttosto chiara.

Il numero in alto (5) corrisponde al numero dei VOB. Se lo stream è MPEG-1 sarà sempre 1.

Successivamente vengono riassunte sia le impostazioni dell'utente che alcune caratteristiche dello stream (*Stream\_Type*, *MPEG\_Type*, *Aspect\_Ratio*, *Picture\_Size*, *Frame\_Rate*, *Location*)

Infine, c'è una serie di cifre apparentemente indecifrabili.

Alcuni di questi valori sono esadecimali, da convertire in binario perchè ogni bit corrisponde all'attivazione di un flag.

- I primi 5 valori sono:
  - **info(Hex):** insieme di informazioni relative ai GOP. Sono 11 bit di cui i primi 8 da destra sono riservati.
  - **matrix:** altri non è che il ColorMatrix
  - **file:** è l'indice del VOB. Per  $n$  VOB, file va da 0 a  $n-1$ .
  - **position-vts-cell:** è la terna che indica la posizione dei frames da decodificare nel DVD.

file, vts, cell, ovviamente, hanno senso solo per stream MPEG-2. Su MPEG-1 avrò **file** = 0 e **vts,cell** = 1

Questo è il **vero** indice che DGindex costruisce per l'accesso ai frames. Tutto il resto sono informazioni prelevate dagli header che descrivono la natura del frame.

Consideriamo la prima riga.

Nell'esempio, **info** corrisponde a *d00*, per cui avrei in binario: **110100000000**.

Ricordo che i primi 7 bit significativi sono riservati. Ci interessano solo gli ultimi 4: **1101**

Da destra verso sinistra:

- **8° bit:** vale 1 se l'immagine I apre un nuovo GOP.
- **9° bit:** vale 1 se le immagini della linea fanno parte di una sequenza progressiva. 0 altrimenti
- **10° bit:** vale 1 se le immagini della linea fanno parte di un GOP chiuso. 0 altrimenti
- **11° bit:** vale sempre 1 per indicare l'inizio di una nuova linea dati

Ora **info** si legge molto meglio. Nell'esempio, vi è una sequenza di *d00*, *900*, vuol dire che ogni riga è un nuovo GOP. Lo ritroveremo più avanti.

- I rimanenti valori sono relativi ai frames, decodificati, uno per uno.

Sempre in relazione all'esempio di prima, prendiamo in esame l'ultimo numero della prima riga, e2 che in binario diventa:

**(11)(10)(00)(10)**

Come prima, in blu sono segnati i bit riservati. In rosso, sono i bit che contengono i flags.

Da destra verso sinistra:

- **1° e 2° bit:** rispettivamente REPEAT\_FIRST\_FIELD e TOP\_FIELD\_FIRST (*RFF* e *TFF* flags). Sono i flags che si utilizzano per capire se c'è un 3:2 pulldown
- **5° e 6° bit:** PICTURE\_CODING\_TYPE flags. In parole povere dice se un frame è di natura I, P, B.
  - 00 (Riservati)
  - 01 (I-Frame)
  - 10 (P-Frame)
  - 11 (B-Frame)
- **7° bit:** PROGRESSIVE\_FRAME flag, vale se il frame è codificato come *interlaced*, 1 come *progressive*. Vedi anche par. 4.1
- **8° bit:** vale 1 se il frame è decodificabile in modo indipendente dal GOP precedente. 0 altrimenti.

Se siete arrivati a questo punto, siete in grado di decodificare ogni singola voce del file *.d2v*.

Certo, fare un'analisi considerando ogni singola voce (anche in un campione) per esaminare il comportamento dei flag più importanti come TFF, RFF, PF è pressochè impossibile.

Un escamotage valido solo per i flags TFF, RFF si ottiene considerando solo l'ultima cifra grazie alla quale si può risalire al pattern usato.

Nel nostro esempio abbiamo una sequenza costante di 2 che corrisponde all'assenza di pulldown (vedi anche **par. 6.3. di "Editing non lineare con Avisynth"**).

Per il video *interlaced* o *progressive* si può ricorrere all'anteprima di DGIndex (che non fa altro che usare

questi valori) ma per tirar fuori le informazioni più importanti si consiglia un altro approccio, disponibile nel menu **Tools** di DGIndex.

Si tratta di

### **4.3. "Parse D2V", analisi del file .d2v**

Parse D2V effettua una scansione del file .d2v e restituisce un file di testo avente lo stesso nome del file d2v e il cui contenuto è la versione compattata di alcune importanti informazioni presenti nel file d2v d'origine, senza che ci sia bisogno di esaminare i flags bit per bit.

Ecco un esempio del file ottenuto:

D2V Parse Output

#### **CODE**

```
Encoded Frame: Display Frames....Flags Byte (* means in 3:2 pattern)
```

```
-----  
[Field Operation None, using flags]
```

```
[GOP]
```

```
[Clip is TFF]
```

```
0 [I]: 0,0.....2
```

```
1 [B]: 1,1.....2
```

```
2 [P]: 2,2.....2
```

```
3 [B]: 3,3.....2
```

```
4 [B]: 4,4.....2
```

```
5 [P]: 5,5.....2
```

```
6 [B]: 6,6.....2
```

```
7 [B]: 7,7.....2
```

```
8 [P]: 8,8.....2
```

```
9 [B]: 9,9.....2
```

```
10 [B]: 10,10.....2
```

```
11 [P]: 11,11.....2
```

```
[GOP]
```

```
12 [B]: 12,12.....2
```

```
13 [B]: 13,13.....2
```

```
14 [I]: 14,14.....2
```

```
15 [B]: 15,15.....2
```

```
16 [B]: 16,16.....2
```

```
....
```

Le informazioni sui flags che prima si reperivano con tanta fatica ora sono immediatamente disponibili.

Il file riporta nell'ordine:

- l'ordine dei frame (TFF o BFF)
- L'inizio e la fine del GOP
- l'indice del frame
- il tipo (I, P, B)

- la coppia di indici dei frames presenti nei 2 fields
- il pattern rappresentato dalla coppia di flags (TFF, RFF). Il numero è decimale e può assumere come valore 0,1,2,3 che rappresentano tutte le possibili combinazioni binarie dei flags, rispettivamente: (vedi sempre **par. 6.3. di "Editing non lineare con Avisynth"**).

In questo file è subito evidente come:

- non vi sia nessun pattern particolare (è una sequenza costante di 2), o meglio, vi è il 2:2 che non provoca nessuna ripetizione di campi.
- i fields contengano l'intera picture, quindi non vi è nessun interlacciamento.

Supponiamo di aver fra le mani un video che ha subito un telecine.

Fermo restando il settaggio di Field Operation al fine di ottenere un file d2v corretto, l'ultima colonna del file restituito da Parse D2V consisterebbe in una sequenza simile a: **3,0,1,2,3,0,1,2....**

L'ordine delle cifre non è vincolante, lo è la ripetizione della sequenza di 4 cifre ed è sufficiente per confermare la presenza del 3:2 pulldown (**cfr. 6.3. di "Editing non lineare con Avisynth"**).

In questo caso già dall'anteprima di DGIndex deve risultare una percentuale **FILM** senza dubbio superiore al 95%

Insomma, in due secondi si riescono a trarre delle conclusioni importanti, difficilmente ottenibili esaminando "manualmente" il file *.d2v*.

In uno stream MPEG-2 un'analisi di questo tipo ha senso.

In uno stream MPEG-1, non avendo i flags tipici dell'MPEG-2, il risultato prodotto da Parse D2V non è di alcun aiuto. L'unico supporto fornito da DGIndex in questo caso è la possibilità di disporre di un indice sulla stream d'origine.

## CONCLUSIONI

DGIndex, come Avisynth, è un piccolo miracolo software. Un tool molto semplice da usare (la creazione di un progetto non richiede mediamente più di 7-8 minuti, compresa l'analisi), indispensabile se siete fra quelli che vogliono tenere sotto controllo ogni aspetto della codifica.

Agli ideatori di questi software va la mia riconoscenza, senza di loro non avrei potuto scrivere niente di tutto questo.

Allo staff di Divxmania va la mia solita riconoscenza per l'aiuto, il supporto e la pazienza. Un giorno, forse, riuscirò a debellare la mia logorrea 😊

**...:Aytin:...**

[Cap. 1. Cos'è DGindex](#)

[Cap. 2. Esplorando i menù](#)

[Cap. 3. DGDecode: plugin per Avisynth](#)