



Ho recentemente ultimato il progetto di un corso su due CD-ROM che spiega l'utilizzo del software di compressione *Cleaner 5.1* della Discreet (precedentemente della Media 100). *Cleaner* è uno strumento potente, e questa è un'opinione che avrei anche se non avessi mai lavorato per la società che lo sviluppa. Ma, con mia sorpresa, ho

damente le strade che mi consentono di aggirare i problemi, ottenendo risultati migliori.

In quest'articolo, spiegherò come ho realizzato il video e le animazioni per i CD-ROM e il sito Web promozionale. I trucchi che ho utilizzato per risolvere i miei problemi vi aiuteranno a risolvere quelli che incontrerete nei vostri progetti. Ma, innanzitutto, un piccolo intermezzo pubblicitario. I miei CD-

mentre ci lavoro e un'interfaccia di navigazione scritta in *Director* (www.macromedia.com). La prima metà del disco esplora alcune delle funzioni più nuove e complesse di *Cleaner 5.1*, come il telecinema inverso e l'integrazione di *Sorenson 3.1 Pro*. La seconda metà include due lunghi tutorial che spiegano lo sviluppo di due progetti dai file sorgente fino ai media compressi definitivi (sul disco sono inclusi

**NOZIONI UTILI
E CONSIGLI CHE
VI EVITERANNO
STRANE SORPRESE
DURANTE
LA COMPRESSIONE
DEI MATERIALI
GRAFICI, AUDIO
E VIDEO PER
LA PRODUZIONE
DI UN CD-ROM E DI
FILE PER IL WEB**

Comprimere video e animazioni per produrre un CD-ROM

dovuto utilizzare molte altre applicazioni per preparare l'audio e il video per questi dischi.

Sono molto esperto nel campo della compressione, ma devo costantemente inventarmi soluzioni ai problemi sempre nuovi in cui incorro quando lavoro col video compresso. Ogni progetto è una sfida. Col maturare dell'esperienza, tuttavia, imparo a vedere più rapi-

ROM di training per *Cleaner* sono disponibili a 189,95 dollari (circa 200 euro) e sono pubblicati dalla Safer Seas. Si possono trovare informazioni aggiuntive sul sito Web della società: www.saferseas.com.

Ho lavorato con media semplici: istruzioni video per ogni sezione, catture animate dello schermo che mostrano l'interfaccia del programma

anche i file sorgente in modo che l'utente possa seguire con precisione il tutorial). Ho creato i video per i tutorial, e il produttore della Safer Seas Mark Kesserman ha montato tutti i contributi dal vivo e ha realizzato la programmazione in *Director*.

Prima che il video venisse montato, ho scritto la sceneggiatura definitiva e ho registrato la mia voce per i tutorial



presso lo studio Digital One (www.digone.com) di Portland (OR, USA), con l'aiuto tecnico di Michael Carter. Carter e Kasserman hanno ammorbidito e rallentato l'audio. Io parlo rapidamente, così hanno rallentato l'audio dell'otto per cento e hanno inserito delle pause per semplificare la sincronizzazione del mio parlato con quel che accadeva sullo schermo e migliorare la comprensione da parte dell'utente. Alla fine, avevamo un totale di circa due ore e mezza di audio. Ho aperto le tracce modificate in *Peak* della BIAS (www.bias-inc.com), le ho poi tagliate

comprimerli nel modo più leggero possibile per farli combaciare con le dimensioni di quelli che gli utenti vedono nei tutorial.

Per il video dei tutorial veri e propri, abbiamo deciso di usare il codec *Animation* di *QuickTime*, che è in grado di eseguire la codifica lossless di lunghe clip. Il codec *Animation* offre ottime prestazioni di riproduzione sui computer moderni. Tuttavia, utilizza un algoritmo molto semplice per la codifica dell'intera clip. Di conseguenza, per ottenere dei keyframe compatti, occorrono lunghe sequenze orizzontali che

mantengano il medesimo valore di pixel. Per ottenere dei delta frame compatti occorrono lunghe linee orizzontali identiche a quelle del fotogramma precedente. Con il codec *Animation*, una schermata di testo o una finestra d'impostazioni di *Cleaner* vengono compresse molto bene, così come soggetti con animazioni semplici come una schermata con il cursore del mouse che si muove sull'interfaccia. Ma ogni volta che si vede la finestra di anteprima a colori del video di *Cleaner*, oppure ogni volta che buona parte dell'immagine a schermo cambia da fotogramma a fotogramma, il data rate sale vertiginosamente. Questo codec non consente un controllo preciso del data rate, si alza o abbassa

re era d'impostare il codec in modalità lossless, con gli slider per la qualità spaziale e temporale impostati a 100, il massimo possibile, e realizzare il video in modo che le cose non complicassero troppo la vita al codec. Abbiamo registrato l'animazione a 640 x 480 pixel e 5 fps, un valore estremamente basso per un video convenzionale, ma sufficiente per vedere il movimento del cursore e mantenere basso il data rate. Sapendo a 10 fps quest'ultimo si raddoppiava o quasi. Abbiamo impostato un keyframe ogni 50 fotogrammi dal momento che i keyframe del codec *Animation* sono molto più grandi dei delta frame. Usare più keyframe avrebbe significato spingere troppo i nostri data rate, mentre inserirne meno avrebbe reso più difficoltoso l'accesso casuale. Per la versione Mac, ho registrato l'attività a schermo con la vecchia e difficile da reperire utility *Cameraman* della Motion Works. Per le mie future registrazioni su Mac userò l'eccellente *Snapz Pro* della Ambrosia Software (www.ambrosiasw.com), dal momento che è migliore di *Cameraman* ed è disponibile sia per Mac OS 9 sia per OS X. Entrambi i programmi sono in grado di registrare direttamente dallo schermo del Mac in un file *QuickTime* col codec *Animation*. Non esistono buone applicazioni per la registrazione dell'attività a schermo che lavorino direttamente in *QuickTime* per Windows, così ho utilizzato *Camtasia Producer* della Tech Smith (www.techsmith.com) che registra su file AVI. Ho convertito in seguito questi file in formato *QuickTime*.

Registrare le animazioni a schermo è stata la parte più dispendiosa in termini di tempo dell'intera produzione. Per



◀ Il secondo disco dei tutorial per *Cleaner* su due CD-ROM di Ben Waggoner, pubblicato dalla Safer Seas (www.saferseas.com), include più di due ore e mezza di video e animazioni. Per comprimere i contenuti, è occorso ben altro che il solo *Cleaner*



in sezioni più brevi per sincronizzarle alle registrazioni dello schermo.

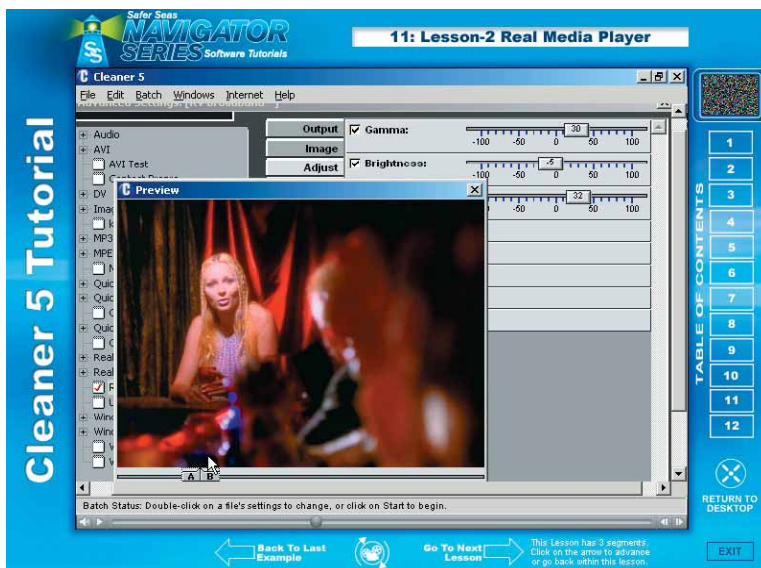
Abbiamo girato inoltre delle introduzioni con me in primo piano per ogni tutorial. Kasserman ha montato e finalizzato queste introduzioni con una DVStorm della Canopus (www.canopuscorp.com). Abbiamo voluto mantenere il codice di *Director* il più simile possibile nelle versioni Mac e Windows del titolo, utilizzando lo stesso formato dei file per entrambe. *QuickTime* resta la scelta migliore per progetti multipiattaforma su CD-ROM, ed è stato utilizzato negli altri progetti della Safer Seas, quindi abbiamo deciso di codificare il nostro video con *QuickTime*.

► Buona parte del video sui dischi mostra *Cleaner* in azione. Il frame rate per queste sezioni è stato impostato a 5 fps. Dal momento che non c'è molto movimento sullo schermo, il frame rate conserva lo spazio su disco e l'immagine non risulta troppo scattosa

La scelta del codec

Dal momento che abbiamo pianificato di utilizzare un CD-ROM da 700 MB, potevamo comprimere tutto il video a circa 75 Kbps (kilobyte al secondo) se non avessimo desiderato mettere nient'altro sul disco. Tuttavia, avevamo bisogno di spazio per l'applicativo generato da *Director*, l'installer di *QuickTime* e i file d'esempio per i tutorial. Per risparmiare spazio, abbiamo scelto di codificare i file d'esempio in formato MPEG-2, ma dovevamo

utilizzare un codec più semplice come una schermata con il cursore del mouse che si muove sull'interfaccia. Ma ogni volta che si vede la finestra di anteprima a colori del video di *Cleaner*, oppure ogni volta che buona parte dell'immagine a schermo cambia da fotogramma a fotogramma, il data rate sale vertiginosamente. Questo codec non consente un controllo preciso del data rate, si alza o abbassa



a seconda della complessità del video. Nonostante il codec offra un controllo spaziale e temporale, questi parametri funzionano semplicemente appiattendoli i colori per rendere le lunghe linee di pixel orizzontali identiche. Il che danneggia sensibilmente la qualità quando la sorgente video viene visualizzata.

Abbiamo deciso che l'ipotesi miglio-

ciascuna clip, preparavo l'audio con la narrazione su un computer. In seguito, su un'altra macchina, impostavo *Cleaner* nel punto esatto di partenza della clip video, premevo il tasto Rec dell'applicazione di cattura dello schermo, attendendo alcuni secondi fino a che non vedevo lampeggiare la luce dell'hard disk sul computer in modo da



sapere che la registrazione era iniziata, e premevo Play sull'altro computer. A questo punto, "recitavo" col mouse seguendo le indicazioni per sincronizzare quel che facevo in *Cleaner* con la narrazione in voice-over. I file audio erano diversi, con durate che andavano da alcuni secondi a svariati minuti, e ogni errore durante una ripresa significava rifare da capo tutta la sezione di video interessata. Ho scoperto di essere un attore terribile sia sul palco, sia sul computer. La registrazione di molte clip ha richiesto diverse riprese. Per completare tutta questa parte del lavoro sono occorse due lunghe settimane.

Terminate le due settimane, la parte relativa ai file per Mac era conclusa. Ma dovevo ancora convertire i file di *Camtasia* su *Windows*. Ho cercato di farlo con *Cleaner*, ma il programma continuava ad andare in crash. *Camtasia* include un'utility di conversione dei file, ma questa rovinava i primi fotogrammi di ogni file, richiedendo il trimming manuale in *QuickTime Player Pro*. Non è un processo complicato, ma l'ho dovuto fare per ciascun file.

► Per catturare le sessioni di *Cleaner* su *Windows*, Waggoner ha usato *Camtasia Producer* della Tech Smith (www.tech-smith.com) per realizzare file AVI dell'attività a schermo. Gli AVI sono stati poi convertiti in file *QuickTime*

Encoding dell'audio

La fase successiva è stata la codifica delle tracce audio. Il mio progetto originale era di utilizzare l'encoding in batch di *Cleaner*. Tuttavia, un bug nel supporto degli MP3 in *Cleaner 5.1* tagliava l'ultima frazione di secondo di ogni file, spesso escludendo l'ultima parola pronunciata. Come scappatoia, ho provato a codificare i file come filmati di *QuickTime* con l'audio in MP3, ma così la seconda metà di ciascun file audio veniva tagliata. Dopo aver perso un'intera giornata a causa di questi bug, ho utilizzato l'utility gratuita *iTunes* su Mac per codificare i file a 48 Kbps (kilobit al secondo), 44,1 KHz mono in formato MP3. L'MP3 comprime senza problemi le voci registrate in modo pulito, e i file risultanti sono buoni quasi quanto l'o-

riginale.

Coi file audio e video individuali completati, il passaggio successivo è stato unirli. Utilizzando *QuickTime Player Pro*, ho aperto ciascun file audio, copiato i suoi contenuti, aperto il file video corrispondente e selezionato Add per incollarci la traccia audio. Una volta salvato il tutto, avevo a disposizione un nuovo file che conteneva sia il video sia l'audio in versione defi-

nit non è in grado di renderizzare il video progressivo, così abbiamo applicato un filtro per il deinterlacciamento adattivo in *Cleaner* per migliorare la qualità delle porzioni statiche del video. Abbiamo inoltre applicato delle impostazioni diverse della gamma per le versioni Mac e *Windows* del CD per ottimizzare la riproduzione su entrambe le piattaforme.

Volevamo il playback a 640 x 480 in



nitiva.

In alcuni casi, mi trovavo con un file audio che finiva poco dopo quello video. Non avevo intenzione di variare la velocità dell'audio o di ricomprimere il video. Così, invece di usare il semplice comando "Add to place", ho utilizzato il comando Add Scaled di *QuickTime* in versione *Pro*. Questo riscalda temporaneamente il video aumentandone leggermente il frame rate per farlo combaciare alla durata dell'audio, un ottimo esempio della potenza di *QuickTime* per questo tipo di lavori.

Il passo successivo è stato preparare i file per le introduzioni video. *StormE-*

modo che gli utenti che lavorano con i file dimostrativi vedano la stessa dimensione della finestra che utilizzo nel video del tutorial. Abbiamo compresso alcuni file di test con *Sorenson Video Pro 3.1*, ma abbiamo trovato che il *Player* di *Director* e *Sorenson Video* richiedevano una maggiore potenza della CPU rispetto al sistema che ci eravamo preposti come requisito minimo per la visualizzazione dei CD-ROM. Abbiamo deciso di usare il codec di *QuickTime VP3* della On2 (www.on2.com) dal momento che il suo decoder non fa un utilizzo altrettanto intensivo della CPU. Il codec *VP3* è parte del

UTILIZZARE LA CONVERSIONE A 8 BIT PER CANALE

► Quando l'immagine sorgente RGB a 24 bit (A) viene convertita in un'immagine a 8 bit con la palette di default di *QuickTime* e il dithering attivo (B), l'immagine appare peggiore rispetto a quando è convertita con una palette 8 bit personalizzata senza alcun dithering (C)



Riducendo il numero di colori dalle milioni di possibilità di una sorgente RGB a 24 bit a una palette di soli 256 colori specifici, si può ridurre notevolmente le dimensioni di un file video compresso. Ma per ottenere risultati ottimali, si deve selezionare con precisione la palette di 256 colori. Il software che avete a disposizione dovrebbe essere in grado di generare una palette coi colori ottimali per i contenuti specifici che si stanno utilizzando. Una palette di default a 256 colori non sarà sufficiente.

La maggior parte dei software hanno come opzione di default l'applicazione del dithering al video mentre lo convertono da colori a 24 bit in colori a 8 bit, causando a un'area composta da una tinta piatta di apparire composta da pattern di colori diversi. Questo può andare bene con sorgenti fotografiche, ma appare terribile con catture da schermo e comporta file compressi con dimensioni maggiori. Per i migliori risultati, si deve portare l'immagine a 256 colori specifici senza utilizzare alcun dithering.

QuickTime Component Download così molti utenti lo troveranno già installato. Ma per essere più sicuri, abbiamo incluso l'installer del codec sul CD-ROM. Abbiamo impostato il codec VP3 in modo che non perdesse fotogrammi o utilizzasse la sua modalità Fast Compress, molto più veloce ma con una qualità leggermente inferiore. Il codec ha prodotto del video di qualità eccellente, anche se alcuni elemen-

KHz, mentre i file di *Media 100* a 640 x 480 con audio a 44,1 KHz. Nei miei test, *Cleaner 5.1* andava continuamente in crash aprendo i file MPEG-2 con l'audio a 48 KHz, così li ho codificati nuovamente a 44,1 KHz per aggirare il problema.

Questo conclude tutta la lavorazione necessaria per la realizzazione del CD-ROM.

in questo caso *Cleaner 5.1* ha prodotto delle immagini corrotte. Allora ho utilizzato il caro vecchio *QuickTime Player Pro* per esportare i file *Animation* come file *QuickTime* compresso con il codec PNG lossless, in cui tutti i fotogrammi sono keyframe. Questo ha prodotto un duplicato "pixel per pixel" della sorgente che *DeBabelizer* era in grado di leggere. La clip che abbiamo selezionato aveva nove sezioni distinte, ma un singolo file di *QuickTime* può contenere sezioni multiple di video, ciascuna con la sua palette, così ho lasciato momentaneamente il video suddiviso in nove file.

Volevamo ottenere una palette ottimizzata a 8 bit per ciascuna sezione, per poi adattare ognuna di esse alla relativa palette. Sfortunatamente, *DeBabelizer* non prevede alcuna funzione per questo. Invece, ho realizzato uno script per *DeBabelizer* che eseguiva un semplice processo. Univa due delle funzioni di batch animation del programma, *SuperPalette* e *Save*, in modo che *DeBabelizer* prima generasse la palette e poi la applicasse al filmato. *SuperPalette* apre ciascun fotogramma di un filmato e genera una nuova palette con il migliore set di 256 colori per riprodurre il filmato. Con solo 256 colori a disposizione, il processo è basato sull'approssimazione. Per il video che mostra un'interfaccia del software che utilizza solo alcuni colori, è facile ottenere una buona palette a 256 colori. Tuttavia, per il video che mostra immagini più naturali, la *SuperPalette* deve rinunciare a molti di essi. La clip



Preparazione per il Web

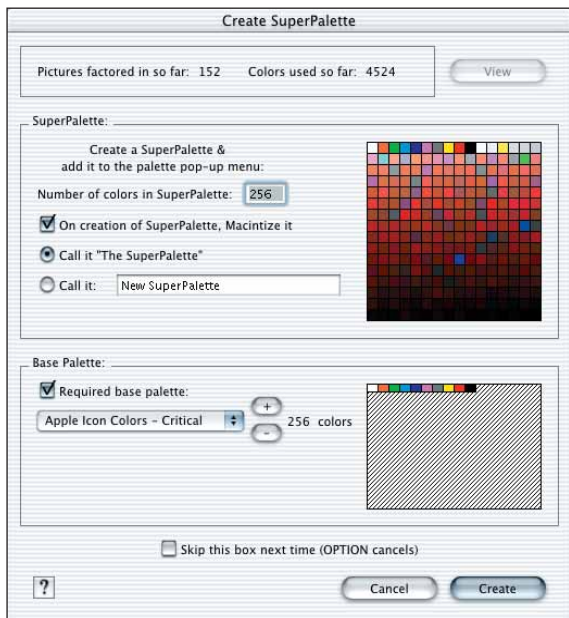
Con l'avvicinarsi della data di consegna, abbiamo deciso di aggiungere dei video d'esempio sul sito Web della Safer Seas. I file del CD-ROM non potevano andare bene, dal momento che anche un breve segmento avrebbe occupato 40 MB. Per il Web, dovevamo codificare i file in modo diverso.

Siamo rimasti fedeli a *QuickTime* come formato di riproduzione,

ma abbiamo scelto un altro codec. Anche in questo caso, i contenuti ci hanno portato alla scelta del codec. Dal momento che la maggior parte dei video relativi ai tutorial mostrano un software che funziona su un computer, non abbiamo selezionato un codec orientato all'ottimizzazione delle immagini naturali. Il codec *Graphics* di *QuickTime* è lossless come *Animation*, ma offre il doppio della compressione. Tuttavia, utilizza una palette a 8 bit, proponendo solo 256 colori. Scegliere la palette giusta è un fattore d'importanza critica. Leggete il riquadro "Utilizzare la conversione a 8 bit per canale" per capirne il perché.

Volevamo pubblicare una sezione di video che fosse interessante e ragionevolmente semplice da comprimere. La parte iniziale del secondo tutorial, con la sua introduzione video e un completo percorso sull'elaborazione precedente alla codifica e le impostazioni di *RealVideo*, era perfetta. Tornando all'era degli 8 bit, lo strumento migliore per la codifica in questo spazio colore era *DeBabelizer* della *Equilibrium* (www.equilibrium.com). Dopo anni trascorsi senza un update, è recentemente stato introdotto *DeBabelizer 5*, quindi ho deciso di mettere alla prova questa nuova release.

DeBabelizer non era in grado di leggere correttamente i file sorgente realizzati col codec *Animation*, apparentemente a causa di qualche problema con le sorgenti compresse con interframe, così dovevo convertire quei file sorgente in un formato adatto. Purtroppo,

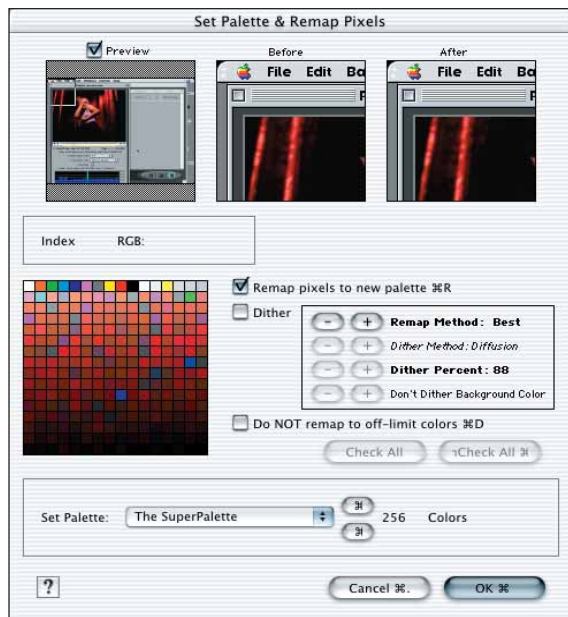


ti, in particolare il testo in movimento, avevano i bordi troppo secchi a causa del deinterlacciamento. Delle problematiche relative alle prestazioni mi hanno costretto a utilizzare l'audio in formato *IMA* mono a 44,1 KHz, che risulta semplice da decodificare per i computer. Dal momento che le introduzioni rappresentavano in totale solo alcuni minuti di video, le dimensioni maggiori non divoravano molto dello spazio a disposizione sul disco.

Essendo a conoscenza delle dimensioni definitive del file video, abbiamo calcolato che restavano a disposizione 210 MB di spazio per i file relativi ai tutorial in formato MPEG-2. È sempre saggio lasciare un po' di spazio in più per le aggiunte dell'ultimo minuto, così abbiamo compresso i file relativi ai tutorial in modo che occupassero 200 MB. Tutte le operazioni di codifica sono state realizzate con *MPEG Power Professional 2.5* della Heuris (www.heuris.com). *Cleaner 5.1* con il suo add-in *Charger* è in grado di produrre video MPEG-2 con una qualità quasi altrettanto buona, ma il software della Heuris ha completato il lavoro in 5 ore anziché in 30. Data la durata complessiva dei nostri due file sorgente, volevamo una velocità di riproduzione di 5.900 Kbps, di conseguenza abbiamo assegnato un data rate di 5.500 Kbps per il video e 384 Kbps per l'audio.

Per far assomigliare il più possibile questi file a quelli che utilizzavo nei tutorial, abbiamo codificato i filmati DV a 720 x 480 pixel con l'audio a 48

◀ *DeBabelizer 5* della *Equilibrium* (www.equilibrium.com), recentemente aggiornato dopo un lungo periodo, ha consentito di ottimizzare alcuni file dimostrativi per il CD-ROM in una palette con colori a 8 bit per un'anteprima del prodotto sul Web



video che abbiamo scelto includeva del video relativamente monocromatico, così la nostra palette definitiva a 256 colori appariva buona.

Un trucco utile coi colori a 8 bit è includere quelli del sistema operativo nella propria palette. Sul Mac, per esempio, questo consente di far apparire correttamente i colori dei menu Apple in *OS 9*. Questo può ridurre leggermente il numero di colori che *DeBabe-*



lizer può assegnare alla SuperPalette, ma l'applicazione potrebbe aver forzato il programma a scegliere almeno alcuni dei colori di sistema, così il compromesso risulta generalmente buono per offrire agli utenti un'esperienza visiva con la quale si sentano a proprio agio.

Una volta pronta, dovevamo salvare nuovamente il filmato con quella palette. Ho cominciato scrivendo uno script per *DeBabelizer* che applicava la palette attraverso i comandi Set Palette e Remap Pixels. Ho disabilitato del tutto il dithering, in modo che ciascun pixel del video venisse convertito per sfruttare al meglio i 256 colori a sua disposizione. Questo assicurava che le aree con un colore piatto nell'originale restassero tali anche nel file in uscita, anche se il colore non era esattamente uguale. Se avessi attivato il dithering, diversi pixel sorgente dello stesso colore avrebbero potuto essere variati con colori diversi nel file convertito.

In seguito, ho impostato uno script Save per leggere ciascun fotogramma del file sorgente, applicare lo script per la mappatura e salvare il file in uscita come filmato *QuickTime* codificato col codec *Graphics*. Infine, ho realizzato un altro script che ha unificato in un solo processo i passaggi SuperPalette e Save. *DeBabelizer* non supporta in modo semplice le variabili, così ho dovuto cambiare manualmente il nome del file sorgente nello script per ciascun file che veniva elaborato. C'erano solo nove file sorgente per le versioni *Mac OS* e *Windows*, così non ci sono stati grandi problemi.

MP3 in *Cleaner 5.1*. La dimensione dei file era critica, così ho utilizzato *LAME* (*Lame Ain't an MP3 Encoder*, www.mp3dev.org), un encoder di MP3 open-source basato su linea di comando che ho compilato dal codice sorgente sotto *Mac OS X*. Una cosa positiva di *LAME* è che si può codificare l'audio VBR con data rate specifici. Ho

frame rate inferiore di video catturato dallo schermo a 640 x 480 con audio a 20 Kbps. *QuickTime* non ha alcun problema a riprodurre i file con contenuti misti.

Alla fine, avevo due file da 7 MB ciascuno. Non male, ma avrei potuto scendere ulteriormente. I file di *QuickTime* hanno un indice degli elementi che, in un filmato complesso come questo, può rappresentare una bella quantità di dati. *QuickTime* consente di comprimere questo header del filmato, riducendo le dimensioni globali del file. Ancora più importante, le dimensioni si riducono nella parte iniziale del file, riducendo in modo significativo il tempo di partenza della riproduzione per il download progressivo dal Web. Inoltre, si possono salvare i file con l'opzione "Disabile Saving", in modo da

```

/usr/bin/login (tty1)
[localhost:~] waggoner% lame -q0 --abr 20 --resample 11.025 /Users/waggoner/Desktop/Safer\ Seas\ demo\Audio\Tutorial\ 2\ Project.aiff
LAME version 3.92 (http://www.mp3dev.org/)
Resampling: input 44.1 kHz output 11.025 kHz
Using polyphase lowpass filter, transition band: 5379 Hz - 5512 Hz
Encoding /Users/waggoner/Desktop/Safer Seas demo/Audio/Tutorial 2 Project.aiff
to /Users/waggoner/Desktop/Safer Seas demo/Audio/Tutorial 2 Project.aiff..m
p3
Encoding as 11.025 kHz average 20 kbps single-ch MPEG-2.5 Layer III (6.8x) qval=
0
  Frame      | CPU time/estim | REAL time/estim | play/CPU | ETA
683/685 (100%) | 0:04/ 0:04 | 0:09/ 0:09 | 7.7878x | 0:00
8 [ 91] *****3
16 [117] *****3
24 [239] *****3
32 [223] *****3
48 [ 44] ****3
48 [ 93] 3
56 [ 93] 3
64 [ 93] 3
80 [ 93] 3
96 [ 93] 3
112 [ 93] 3
128 [ 93] 3
144 [ 93] 3
160 [ 93] 3
average: 23.4 kbps

Writing LAME Tag...done
[localhost:~] waggoner%

```

utilizzato questo comando per codificare il parlato per le animazioni a schermo:

```
lame -q0 --abr 20 --resample 11.025
```

Il comando si può suddividere in questo modo: utilizza la modalità di encoding più lenta, con la qualità maggiore (-q0), con un bit rate variabile a 20 Kbps di media (--abr 20) e una fre-

quenza di campionamento di 11.025 KHz (--resample 11.025). *LAME* ha prodotto file con un data rate leggermente più alto di quello richiesto, con una media di circa 23 Kbps, ma con una qualità sonora adeguata.

Utilizzando la stessa tecnica usata per i file su CD-ROM, ho utilizzato *QuickTime Player Pro* per aggiungere i file MP3 a quelli video. Così mi sono trovato con dieci file per ciascuna piattaforma: una sezione per il video di apertura e nove sezioni relative ai tutorial. Sempre utilizzando *QuickTime Player Pro* ho incollato tutti i video insieme e ho salvato il file risultante come un filmato unico. Questo mi ha consentito di avere un solo file 640 x 480 con una sezione di video introduttivo a 320 x 240 coi pixel raddoppiati e audio a 48 Kbps, più nove sezioni con

evitare la modifica del file definitivo da parte degli utenti. *Cleaner 5.1* può "appiattire" un file esistente senza ricompressione mentre genera un header compresso per il filmato e applica il comando *Disable Saving*. Tuttavia, un bug di *Cleaner 5.1* reseta la risoluzione del video d'apertura raddoppiato e lo riproduce a 320 x 240 anziché a 640 x 480, annullando l'effetto di duplicazione dei pixel.

L'inestimabile *LiveStage Pro* della *Totally Hip* (www.totallyhip.com) è stato usato per comprimere il complesso header per i filmati d'esempio da pubblicare sul Web

LiveStage Pro della *Totally Hip* (www.totallyhip.com) è accorso in mio salvataggio. Nonostante la mia esigenza risultasse piuttosto basilica per questo raffinato software di

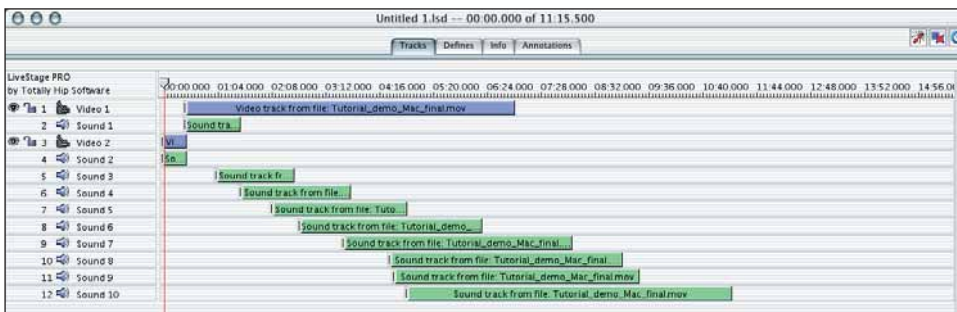
authoring interattivo per *QuickTime*, *LiveStage* ha importato perfettamente il mio filmato *QuickTime* e lo ha esportato con l'header compresso e il salvataggio disabilitato.

Ecco fatto. I video sul Web offrono un ottimo assaggio del progetto in un file scaricabile con un qualsiasi modem. Il CD-ROM sta vendendo bene. Tuttavia, la natura paradossale del progetto non mi è sfuggita. Nonostante *Cleaner* resti uno strumento potente, ho incontrato una serie di bug nella versione 5.1 che mi hanno trattenuto dall'utilizzarlo per la realizzazione dei materiali per i tutorial che riguardavano proprio questo programma. Spero che la *Discreet* (che essendo una divisione della ricca *Autodesk* dispone di maggiori capitali rispetto alla *Media 100*) sia in grado di bonificare le aree malsane di questo software nella prossima release.

(© DV)

CG

► Waggoner ha usato l'encoder open-source *LAME* (www.mp3dev.org) per convertire i contributi audio in MP3 per i file d'esempio sul Web



Conversione delle introduzioni

Il grosso del mio lavoro a questo punto era terminato, ma dovevo ancora convertire i video d'apertura, l'audio e le narrazioni per il Web. Per il video d'apertura, ho realizzato un file in formato *Sorenson Video 3.1* a 320 x 240 pixel di risoluzione impostato per essere riprodotto a 640 x 480. Utilizzando la duplicazione dei pixel si mantengono bassi il data rate e i requisiti per il playback, ma si ottiene un file che può comunque essere riprodotto a risoluzione piena. Dal momento che il video introduttivo includeva della musica, ho codificato l'audio in formato MP3 VBR a 48 Kbps in mono. Il bug che ho menzionato in precedenza non consentiva di codificare i file del parlato in

quenza di campionamento di 11.025 KHz (--resample 11.025). *LAME* ha prodotto file con un data rate leggermente più alto di quello richiesto, con una media di circa 23 Kbps, ma con una qualità sonora adeguata.

Utilizzando la stessa tecnica usata per i file su CD-ROM, ho utilizzato *QuickTime Player Pro* per aggiungere i file MP3 a quelli video. Così mi sono trovato con dieci file per ciascuna piattaforma: una sezione per il video di apertura e nove sezioni relative ai tutorial. Sempre utilizzando *QuickTime Player Pro* ho incollato tutti i video insieme e ho salvato il file risultante come un filmato unico. Questo mi ha consentito di avere un solo file 640 x 480 con una sezione di video introduttivo a 320 x 240 coi pixel raddoppiati e audio a 48 Kbps, più nove sezioni con