

P... come Pixel

Testo e fotoelaborazioni di Andrea de Prisco



Come certamente già sapete, per festeggiare i primi 150 numeri della nostra amata rivista (il «complinúmero» è stato lo scorso mese di Aprile), abbiamo deciso di realizzare un CD-ROM contenente una raccolta selezionata dei migliori articoli pubblicati da MC in questi 14 anni di informatica personale.

Andando, dunque, a rimettere il naso negli articoli che scrivevo una decina di anni fa, mi sono accorto come a quei tempi fosse dominante nei miei «pezzi» più il lato didattico-teorico che quello pratico-esecutivo. Devo, tra l'altro, buona parte della mia «evoluzione giornalistica» proprio ad una frase di Marco Marinacci rivoltami agli inizi degli anni Ottanta. Suonava più o meno così: «L'importante è farsi capire dai lettori. Nei passaggi più difficili è meglio spiegare le cose due volte di seguito, in due modi anche solo leggermente diversi, in modo da esser sempre piuttosto certi di non divulgare fischi per fiaschi».

In quei tempi, e avendo fatto tesoro delle raccomandazioni del capo, riuscivo a portare sulle pagine di MC temi assolutamente impensabili per una rivista mensile di informatica, come la teoria della computabilità logica, la struttura

dei sistemi di calcolo (dal livello microcodice dei processori fino allo sviluppo di un protocollo di rete fault tolerant denominato ADPnetwork in onore della mia... infinita modestia) o alcune estensioni dei linguaggi di programmazione disponibili negli «home computer» dell'epoca.

In questa serie di articoli dedicati all'imaging digitale, di teoria effettivamente ne abbiamo fatta ben poca. L'approccio, per quanto possibile, è sempre stato quello del «problema-risoluzione» e siamo andati avanti in questo modo mostrandovi continuamente esempi pratici.

Mi è così venuta voglia, questo mese, di tirare (momentaneamente) il freno a mano, per tuffarci brevemente negli aspetti di base della grafica, per essere certi - questa è la mia speranza - che chi mi segue in questo divertente viaggio abbia ben chiari alcuni concetti fondamentali. I prossimi articoli, infatti, non potranno più prescindere da queste nozioni di base, visto che toccheremo via via temi più avanzati (dalla correzione cromatica alla corretta stampa a colori, solo per anticiparvi qualcosa). Mi rendo conto, infatti, che sin dall'inizio ho dato molte cose per scontate, con

buona pace per chi segue MC (o il mondo dell'informatica) ormai da molti anni, ma comunicando «male» con chi è completamente a digiuno di aspetti tecnici o che si è avvicinato solo da poco a questo entusiasmante mondo.

Quindici o venti anni fa, far funzionare un computer era sinonimo di «programmare», da ormai una decina d'anni la stragrande maggioranza degli utenti sono veri e propri «utenti» e non necessariamente «tecnici». Oggi un hard disk, tanto per fare un esempio, è solo un posto dove registriamo le nostre cose (poco più di una videocassetta...), avendo quasi del tutto dimenticato che si tratta di una struttura complessa sia dal punto di vista fisico che logico basata su svariate testine, tracce, settori e (SOPRATTUTTO!) un file-system.

Questo mese parleremo principalmente di pixel (acronimo di Picture Element, ma questo credo che sia del tutto inutile ribadirlo), di risoluzione grafica e risoluzione cromatica. Non vedremo elaborazioni digitali vere e proprie (ad eccezione della sezione dedicata ai lettori) ma sarà comunque una lettura interessante.

O, almeno, spero!

Binari digitali

Secondo alcuni tra i più fantasiosi teorici (perché dar loro torto?) sembrerebbe che noi «umani» utilizziamo per i nostri calcoli le cifre decimali (da 0 a 9) per il semplice fatto che abbiamo dieci dita. Probabilmente se ne avessimo avute otto, i nostri numeri sarebbero stati formati dalle sole cifre da 0 a 7. Tutto questo senza minimamente intaccare la nostra naturale evoluzione, visto che anche otto sole cifre sono comunque sufficienti per rappresentare qualsiasi entità numerica e fare con queste ogni possibile operazione. Molto probabilmente sarebbe stato più contento Pitagora, con la sua tavola di sole 64 caselle in luogo di 100, e naturalmente gli scolaretti costretti ad imparare a memoria solo otto tabelline (per di più di sole otto righe!) invece di dieci. Certo, con sole otto dita dovrebbe essere più difficile suonare gli strumenti musicali, ma non per questo Bach e Chopin ci avrebbero lasciato a bocca asciutta.

Anche con sole otto cifre, infatti, utilizzando i medesimi meccanismi della comune aritmetica decimale (che in questo caso prende il nome di «ottale» o «base 8»), avremmo potuto ugualmente rappresentare tutti i numeri naturali, i numeri reali, i frazionari e così via. Al posto di unità, decine, centinaia e migliaia, avremmo unità, «ottine», «sessantaquattre», «cinquecentododici»... ma il succo non cambia.

A questo punto, la domanda nasce spontanea: con sole otto cifre, come possiamo scrivere le quantità superiori a 7? Semplice la risposta: allo stesso modo in cui nell'aritmetica in base 10 (decimale) riusciamo a scrivere quantità superiori a 9: utilizziamo più cifre!

Ad esempio, la quantità decimale 14 (pari a una decina e quattro unità) in base 8 viene rappresentata da una «ottina» e sei unità (si scrive 16, ma vale sempre quattordici!). Cambiano i termini della scomposizione, ma non il procedimento. E questo vale per qualsiasi numero, quanto grande vogliamo, utilizzando sempre e soltanto le cifre da 0 a 7. Naturalmente esiste anche un procedimento per passare da ottale a decimale ed è facile dimostrare che tale trasformazione è sempre possibile per qualsiasi entità numerica.

Discorso analogo per le operazioni aritmetiche: la somma, la sottrazione, la moltiplicazione e la divisione restano tali e quali, ricordando di utilizzare sempre le giuste tabelline per non diventare scemi.

Tutto questo vale per qualsiasi «base», anche maggiore di 10 (come l'aritmetica esadecimale in base 16) o la minima possibile che si chiama «binaria» (base 2).

Quest'ultima, per motivi strettamente legati alla semplificazione elettronica, è quella utilizzata dai computer o, più in generale, da tutti i dispositivi digitali. Ragionare in base 2 vuol dire utilizzare

due sole cifre, 0 e 1, due sole tabelline cortissime (urrah!) e una tavola pitagorica a dir poco comica: 4 caselle!

Anche in base due è in ogni caso possibile rappresentare qualsiasi numero: certo, utilizzeremo numeri lunghissimi anche per quantità numeriche tutto sommato modeste, ma come abbiamo già detto nulla cambia dal punto di vista della calcolabilità di un'aritmetica di questo tipo.

Una cifra binaria, 0 o 1, è notoriamente un bit (contrazione di binary digit) e generalmente otto bit formano un byte. Tutte le possibili combinazioni di 0 e di 1 che possono stare in un byte sono 256 (non è una quantità «a caso», è pari a 2, la base, elevato ad 8, il numero di cifre in questione), quindi con un solo byte (di otto bit) è possibile rappresentare i primi 256 numeri naturali (da 0 a 255).

Ciò premesso, lasciamo da parte byte e «sessantaquattre» e concentriamo la nostra attenzione sull'immagine digitale.

Picture element

Premesso che un'immagine digitale è formata da una quantità più o meno grande di pixel (maggiore è il loro numero più l'immagine digitale è ricca di informazione e quindi di dettagli), la prima considerazione da fare riguarda il fatto che se le dimensioni di questi elementi sono più piccole della naturale

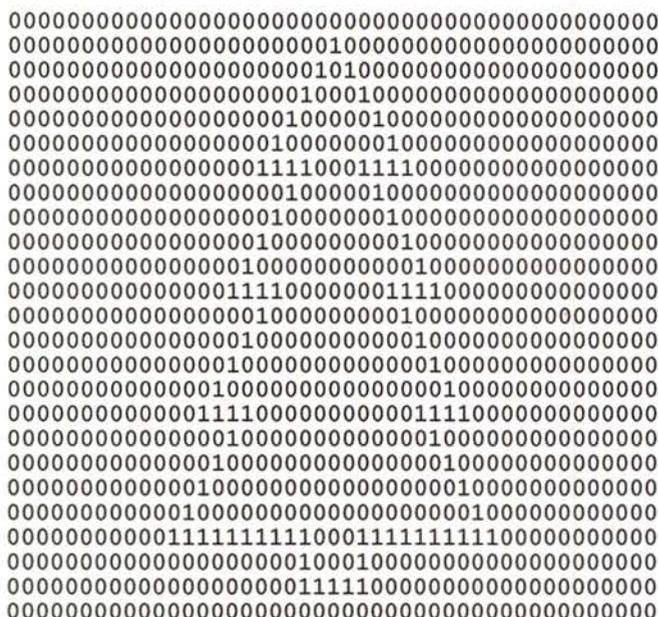


Figura 1a: Un'immagine digitale a bassissima risoluzione in bianco e nero rappresentante un alberello di Natale.

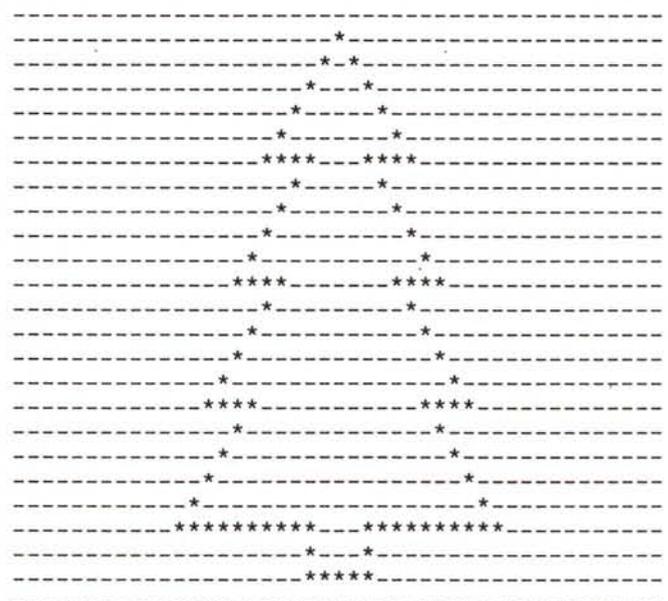
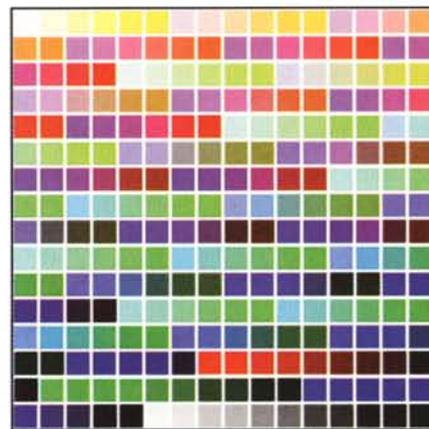


Figura 1b: Lo stesso alberello di figura 1a sostituendo ad ogni 0 un trattino e ad ogni 1 un asterisco.



L'immagine di partenza a 16.7 milioni di colori. Qui a lato è mostrata solo una piccola frazione dell'enorme tavolozza dei colori effettivamente disponibili. È impossibile, infatti, visualizzare quasi diciassette milioni di pixel contemporaneamente per mostrare tutti i colori disponibili.

Secondo caso: l'immagine è stata portata a soli 256 colori, utilizzando la «palette» di sistema: 256 colori fissi e immutabili da utilizzare per le nostre immagini (mostrati qui a lato). La qualità finale, come è visibile, è piuttosto bassa: i colori sono pochi e mal utilizzati.



grana fotografica, nessuna perdita di definizione è imputabile al passaggio da immagine tradizionale alla sua rappresentazione digitale all'interno di un computer. Dato che un'immagine digitale è formata da bit (e quindi da byte) più saranno i pixel di cui è formata maggiore spazio occuperà in memoria.

Un fotogramma in formato 35mm (24x36) è formato da svariati milioni di minuscoli granuli, per una sua codifica senza perdita di dettaglio alcuna sono necessari molti megabyte. Ciò dipende, oltre che dal numero di pixel utilizzati (di quest'aspetto parleremo alla fine), anche dal numero di colori codificabili: in pratica ogni singolo pixel in quanti colori diversi può essere rappresentato.

Rimanendo in termini generali, ad ogni pixel di un'immagine digitale è associato un certo numero di bit. Più bit sono associati ad ogni pixel più colori riusciamo a rappresentare. Se la nostra immagine digitale è composta da soli pixel bianchi e neri (come in una stampa tipografica ad un solo colore) è suffi-

ciente un bit per ogni pixel. Se, sempre ad esempio, decidiamo di associare il valore 0 ad ogni pixel bianco ed il valore 1 ad ogni pixel nero, abbiamo già effettuato la nostra prima, semplice, codifica. In figura 1a è rappresentata (mi raccomando non ridete!) la codifica di un'immagine digitale strettamente bianco/nero di un albero di Natale. Dato che i colori sono solo due (bianco e nero) è sufficiente, come detto, un bit per ogni pixel. In figura 1b, per maggiore chiarezza è mostrata la stessa immagine sostituendo un trattino ad ogni 0 e un asterisco ad ogni 1.

Un'immagine di questo tipo, come è facile verificare, è formata da 1125 pixel (25x45) e dato che ogni pixel è codificato con un solo bit, lo spazio necessario per la sua rappresentazione in memoria è pari a 1125 bit.

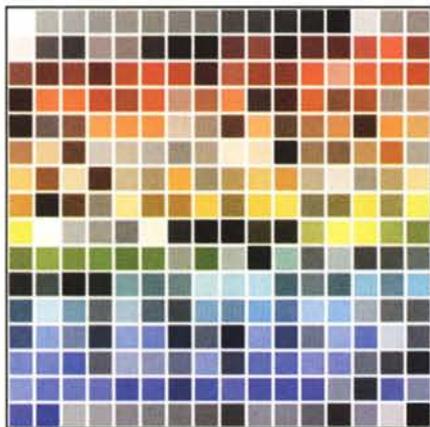
Associando un numero maggiore di bit ad ogni pixel, possiamo digitalizzare della nostra immagine anche le sfumature di grigio o un pari numero di colori. Proseguendo nel nostro esempio, se avessimo associato due bit per ogni pixel avremo avuto come conseguenza la possibilità di codificare (e quindi utilizzare) quattro colori o livelli di grigio. Due bit, infatti, permettono di ottenere quattro distinte combinazioni di 0 e di 1, che possiamo associare ai seguenti colori:

00 —> Bianco
01 —> Grigio chiaro
10 —> Grigio scuro
11 —> Nero

Naturalmente possiamo utilizzare le stesse combinazioni per ottenere altri colori, secondo le nostre necessità. Ad esempio:

00 —> Bianco
01 —> Verde
10 —> Rosso
11 —> Blu

Dovrebbe esser chiaro a questo punto che più bit associamo ad ogni pixel, più colori possiamo codificare e quindi trattare, memorizzare e visualizzare. Come rovescio della medaglia, è altrettanto vero che un maggior numero di colori implica sia una maggiore occupazione in memoria dell'immagine digitale sia una gestione più pesante e laboriosa quando è necessario eseguire un trattamento digitale sulla nostra immagine.



Terzo caso: i colori sono sempre 256 ma la «palette» è stata costruita appositamente dal sistema per «azzeccare» un numero più ampio possibile di colori originali. Il risultato è piuttosto gradevole, pur essendo comunque distante anni luce da quello originario. A lato la «palette» ottimizzata.

Con soli 16 colori, pur ottimizzando la palette, il risultato ottenuto è proprio inaccettabile. Per tentare di ovviare al problema con soli 16 colori si può ricorrere al meccanismo del dithering (vedi testo) ma in questo caso, pur guadagnando qualcosa in termini cromatici ci «rimettiamo» in definizione.



Con 256 colori (8 bit per pixel) il risultato «comincia» ad essere gradevole, specialmente se associamo a questi il meccanismo della «palette». Per non scendere a compromessi cromatici, utilizzando una codifica RGB (basata sui tre colori primari della sintesi additiva), è necessario utilizzare almeno 24 bit (tre byte) per pixel, con i quali possiamo rappresentare più di 16.7 milioni di colori: per la precisione, 256 livelli di rosso, 256 livelli di verde, 256 livelli di blu per ogni punto. Se prendete una calcolatrice e moltiplicate $256 \times 256 \times 256$ ottenete come risultato 16.777.216 che è il numero totale di combinazioni possibili con una codifica di questo tipo.

Grazie al fatto che l'occhio umano difficilmente riesce a notare differenze tra un'immagine a 16.7 milioni di colori e un'immagine reale (notoriamente composta da infiniti colori), tale codifica è soprannominata «true color». Ovviamente si tratta di vero e proprio «falso tecnologico» che sfrutta le nostre limitate (per modo di dire...) capacità per-

cettive. Infatti, per quanto possano sembrare tanti 16.7 milioni di colori, non sono «in pratica» tantissimi una volta constatato che sono la combinazione di 256 livelli di ogni colore primario. È il numero 256 che, in alcuni casi, è spaventosamente piccolo quando cerchiamo di correggere o modificare una ben precisa componente cromatica di un'immagine. Ma di quest'aspetto, lo prometto, parleremo in uno dei prossimi articoli di Digital Imaging.

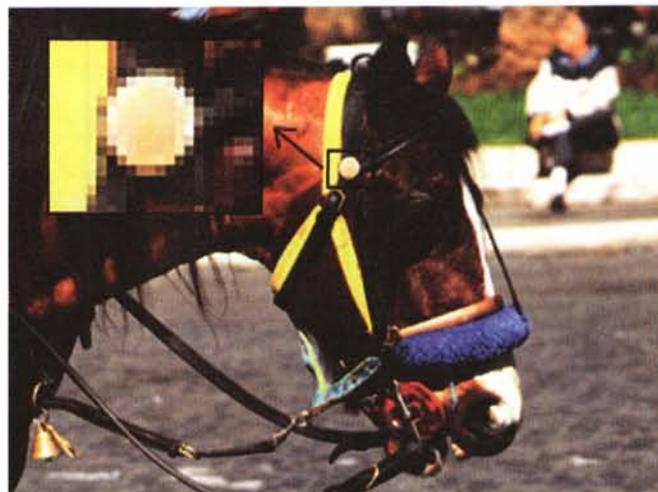
Colori e Palette

Facciamo un passo indietro e, lasciando momentaneamente da parte la codifica «true color», diamo un rapido sguardo ad alcune tecniche particolari con le quali è possibile ottenere, a parità di spazio occupato in memoria, risultati visivamente più accettabili sotto il profilo cromatico.

La prima e più diffusa tecnica consiste nell'utilizzo di una «palette» che permette di ottimizzare l'uso dei colori disponibili il cui numero, come detto, dipende esclusivamente dal numero di bit associati ad ogni pixel. Se il nostro sistema digitale lavora ad esempio con soli 256 colori e questi sono stabiliti in maniera fissa e immutabile dal costruttore una volta per tutte, potremmo

avere un po' di problemi per tutte quelle immagini in cui i colori non corrispondono «più di tanto» ai 256 prefissati. Vedremo le nostre immagini, specialmente se ricche di sfumature, con i colori falsati per il fatto che il sistema ha dovuto utilizzare solo quelli disponibili o, molto spesso, solo un ristretto sottoinsieme di questi. Si può rimediare parzialmente a questo problema attraverso il cosiddetto «dithering» che permette, accostando punti di colore diverso, di ottenere nuove tinte non comprese nell'elenco originario. Ma in questo caso abbiamo una perdita di risoluzione e pur guadagnandoci dal punto di vista cromatico ci rimettiamo in definizione.

Tornando, quindi, al caso dei 256 colori «fissi e immutabili», in pratica vuol dire che disporremo di una decina di grigi, una decina di blu, una decina di rossi, una decina di verdi, una decina di gialli, una decina di marroni e così via (fino a 256 colori totali). Se nella nostra immagine digitale sono presenti solo sfumature di blu e sfumature di verde



In un'immagine ad alta risoluzione (a sinistra), come è ovvio, ci sono molti più dettagli che in una a bassa risoluzione (a destra). Nelle due immagini è stato evidenziato, per comodità, un particolare: la differenza, comunque, si vede anche ad occhio nudo.

(ad esempio la foto di un prato con un bel cielo sullo sfondo impreziosito da qualche nuvola) sebbene il sistema metta a disposizione 256 colori noi riusciremo ad utilizzarne soltanto una ventina, i dieci verdi e i dieci blu.

Per ovviare a questo inconveniente, è sufficiente utilizzare, in luogo dei 256 colori «fissi e immutabili», un elenco di colori ridefinibili dall'utente: una palette. In questo modo i 256 colori, disponibili non sono fissati dal costruttore, ma possono essere impostati dall'utente (o automaticamente dal sistema). Ferma restando quindi l'occupazione di memoria per quel che riguarda l'immagine (8 bit per pixel), ognuna delle possibili 256 combinazioni non

corrisponde più ad un colore preimpostato, ma ad una determinata casella della «Palette»: se ogni casella di quest'ultima è formata da 24 bit, avremo sì 256 colori, ma a scelta tra gli oltre 16.7 milioni codificabili con una tale quantità di informazione. Ritornando all'esempio di prima, possiamo definire oltre un centinaio di sfumature diverse di verde ed altrettante sfumature diverse di rosso (oltre ad una cinquantina di bianchi, grigi e grigetti per le nuvole) per ottenere un'immagine ben più realistica.

Un secondo sistema per diminuire l'occupazione di memoria è denominato HAM (hold and modify) ed è molto utilizzato in ambiente Amiga. Il modo HAM più recente (il nome esatto è

HAM8) utilizza come nel caso precedente 8 bit per ogni pixel, ma vengono interpretati nel seguente modo: se i primi due bit sono posti a 00, i rimanenti sei indicano uno dei sessantaquattro colori definiti in un'apposita palette di tale dimensione. Questi sessantaquattro colori (due elevato a sei equivale per l'appunto a tale quantità) rappresentano i colori di base e sono anch'essi liberamente definibili dall'utente. Viceversa, se i primi due bit sono posti a 01, i rimanenti sei bit indicano la quantità di blu mentre le componenti rosso e verde restano invariate rispetto al pixel precedente. Se i primi due bit sono posti a 10, è il rosso a giocare la sua carta (i rimanenti sei bit indicano il valore per

A tutti i lettori

Se siete interessati all'argomento «Digital Imaging» potete anche voi contribuire alla nostra rubrica inviando in redazione alcune fotografie scattate da voi stessi, delle quali vorreste modificare o correggere alcune componenti, al fine di migliorare il risultato finale. Non inviate, però, foto sfocate o mosse in quanto nulla è possibile fare a riguardo se non riscattare la fotografia con più attenzione la prossima volta. Fate riferimento, se volete un'idea circa la fattibilità, alle immagini che mensilmente pubblichiamo in queste pagine. È importante, in ogni caso, inviare sempre una stampa su carta (anche di formato piccolo) delle vostre immagini e mai (MAI!) gli originali su pellicola negativa o diapositiva. Per i soliti problemi organizzativi, il materiale inviato non verrà restituito.

Ogni mese, la proposta più interessante verrà gratuitamente elaborata presso la nostra redazione e il risultato pubblicato in queste pagine. Per questo motivo è necessario allegare alle fotografie una dichiarazione liberatoria, firmata dall'autore delle fotografie, in cui si dichiara la paternità delle stesse e se ne autorizza la pubblicazione sulle pagine di MCmicrocomputer.

Chi, invece, fosse già attrezzato per effettuare elaborazioni di immagini, può inviare su disco Mac o MS-DOS i propri lavori più interessanti, inserendo sempre (va bene un qualsiasi formato diffuso: PICT, TGA, JPG, TIFF, GIF, PCX, ecc.ecc.) l'immagine originaria, l'immagine elaborata, una breve descrizione dei procedimenti utilizzati e, stampata su carta e sottoscritta, la dichiarazione liberatoria di cui sopra. Ogni mese il lavoro più interessante verrà pubblicato su MCmicrocomputer e l'autore (se non si tratta di un professionista nel campo dell'imaging digitale) ricompensato con un gettone di 100.000 lire. Fatevi avanti!

P... come Polaroid

Le immagini di quest'articolo (escluse quelle dei lettori) sono state digitalizzate con il Polaroid SprintScan 35, provato sul numero 150 di MCmicrocomputer. Si tratta, come abbiamo avuto modo di verificare durante i nostri testi, di un apparecchio di ottima qualità, in grado di digitalizzare in pochi secondi sia negativi che diapositive 35mm.



S.O.S. Digital Imaging

L'immagine scelta questo mese per il fototriccio digitale ci è stata inviata dal lettore Lillo Mancuso di Torino. Si tratta di uno scorcio interno del Palazzo Ducale di Venezia, grandioso edificio civile della Serenissima, tra i più ammirati e celebri palazzi del mondo per la concezione architettonica e i tesori d'arte in esso racchiusi, nonché prestigiosa sede del Doge e delle più alte magistrature della Repubblica veneta.

«*Ammirato quanto deturpato*», come afferma lo stesso autore, «*da quelle orribili transenne, che oltre ad impedirne l'accesso ai piani superiori (l'immagine risale a sette anni fa), non ne consentivano nemmeno una dignitosa ripresa fotografica*».

Eliminare particolari indesiderati, come più volte abbiamo avuto modo di mostrare in queste pagine, non è certo un problema, specialmente quando è possibile attingere ad altre porzioni dell'immagine per ricostruire i dettagli mancanti. Agendo per via digitale, con il consueto strumento «Timbro», le transenne sono scomparse facilmente e in pochi minuti.

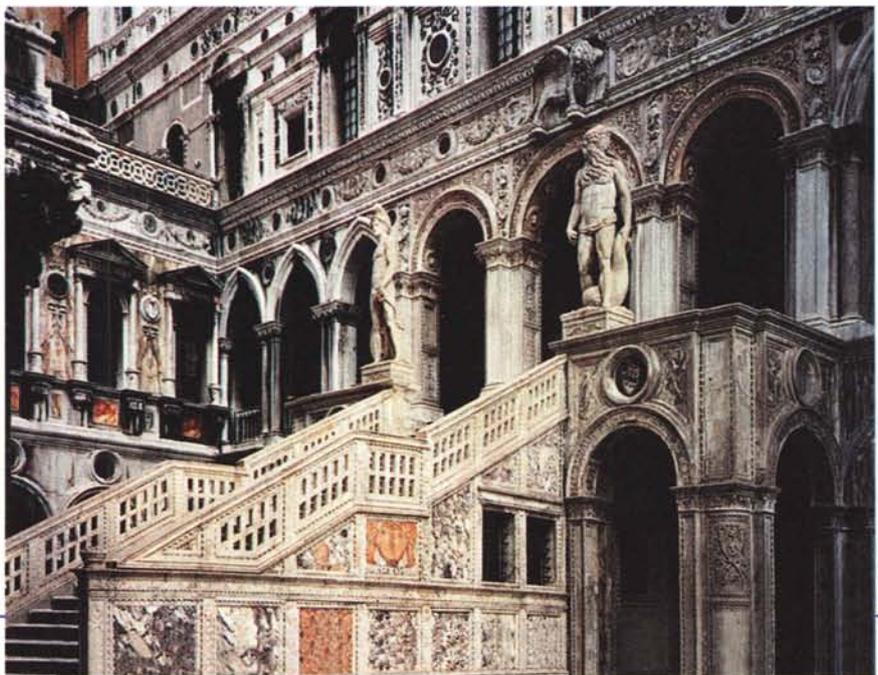
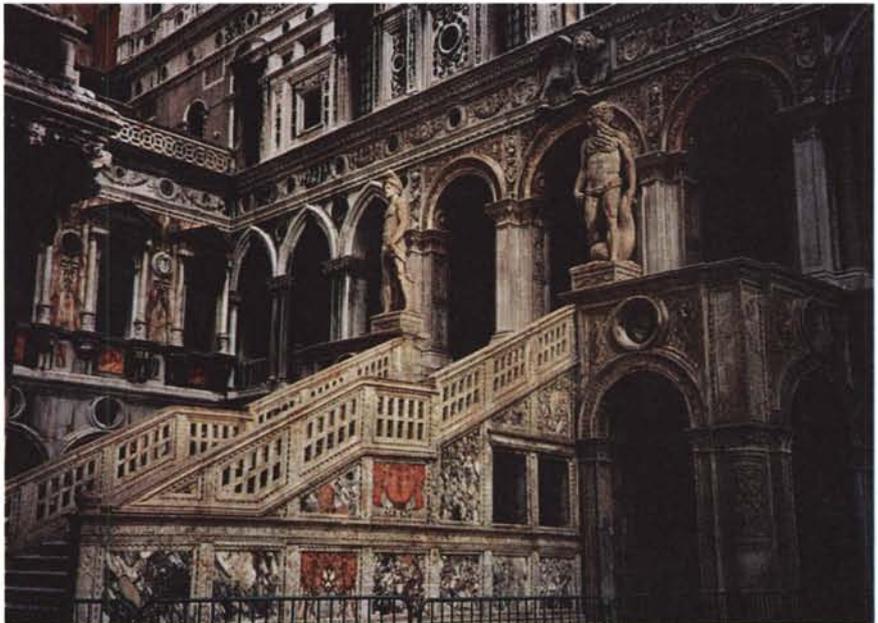
L'immagine originale (qualità della stampa ricevuta a parte: mi chiedo perché certi laboratori continuano ad esistere...) è comunque piuttosto bella. Peccato che un ulteriore elemento di disturbo (non segnalato dall'autore) sia rappresentato dalle linee cadenti, dovute alla ripresa dal basso verso l'alto. Nella fotografia tradizionale, per correggere questo tipo di inconveniente, è necessario disporre di una macchina fotografica a banco ottico o, quanto meno, di un obiettivo decentrabile montato su un apparecchio reflex. Inutile aggiungere che è strettamente necessario un robusto treppiedi, un po' di pazienza (non sono certo fotografo da gita scolastica) e, naturalmente, uno schermo di messa a fuoco quadrettato, fatto apposta per la fotografia architettonica.

A noi, ovviamente, non ce ne frega niente né del basculaggio né del treppiedi (al diavolo pure il fotolaboratorio da strapazzo!) e raddrizziamo digitalmente le linee cadenti non senza un preventivo riequilibrio dei livelli ed una sistematina alla curva di gamma (in pratica gli ho fatto il tagliando dei 120.000 chilometri).

Per annullare la distorsione prospettica è stato utilizzato come sempre Photoshop (Sua Eccellenza!) e in particolare la funzione «Distorsione» presente nel sottomenù Effetti del menu Immagine. La prima operazione da compiere sarà quella di valutare il

livello di intervento relativo alla correzione. Per fare questo si tracciano (al limite sulla fotografia cartacea, io ho utilizzato sull'immagine digitale un livello aggiuntivo gentilmente offerto da Photoshop 3.0) da ognuno degli angoli inferiori due linee nella direzione del punto di fuga della nostra prospettiva da correggere, misurando a che distanza dai rimanenti due angoli intersecano il bordo superiore dell'immagine. A questo punto si seleziona l'intera immagine e, dopo aver richiamato la funzione «Distorsione», si trascinano uno alla volta gli angoli inferiori per una distanza pari a quella misurata precedentemente sul bordo superiore. Effettuato

il posizionamento dei due angoli, possiamo dare l'OK (semplicemente cliccando all'interno dell'area) e in pochi secondi otteniamo la trasformazione voluta. L'immagine appena distorta (nel nostro caso corretta) è mostrata nella fotografia piccola: le linee verticali ora sono tutte parallele, il punto di fuga è stato rispedito a distanza infinita, e basterebbe selezionare una porzione rettangolare interna al trapezio per ottenere il nuovo taglio. Sempre per la serie «Non mi basta mai» (cfr MC 151 pag. 196) ho preferito selezionare una porzione più ampia, ricostruendo, ovvero inventando, i pezzi mancanti. Che ve ne pare?



Il contributo dei lettori

Volare, oh! oh! Elaborare oh! oh! oh! oh!

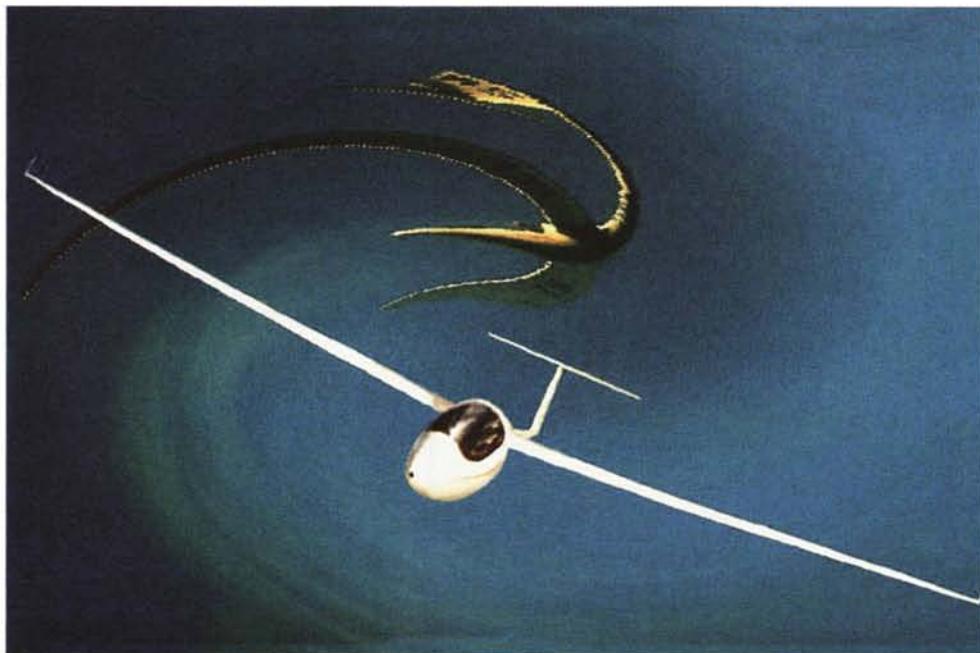
L'immagine scelta questo mese tra tutte quelle giunte in redazione (una vera e propria montagna, complimenti!) ci arriva dal lettore Giancarlo Bresciani di Ferrara, pilota di volo a vela con già duemila ore di volo sulle spalle. Unendo la passione per tale sport a quella per i computer, ha creato l'immagine finale (mostrata in basso a destra) partendo da due fotografie dell'autore, successivamente trasferite su PhotoCD Kodak per una sana «fruizione digitale» delle stesse. Lasciamo la parola al nostro stimato lettore: «... ho trasformato l'aliante della prima foto in una specie di uccello preistorico. In primo piano, invece, si palesa aggressivo un altro aliante, ricavato dalla seconda foto. Ho inteso, in questo modo, lanciare due messaggi: nel primo si vuol far credere che nell'archeopteryx fosse già nascosto un aliante, nel secondo vi è il passaggio dall'essere più antico in gra-

do di volare al più moderno, sintetizzando metaforicamente tutta la storia del volo...».



L'immagine è stata scelta per tutta una serie di motivi. Innanzitutto quello estetico: si tratta di un giudizio certamente soggettivo, ma proprio da questo punto di vista mi è piaciuta moltissimo. In secondo luogo è il frutto combinato di diverse tecniche, tutte sapientemente sfruttate. Rappresenta, infatti, una chiara dimostrazione del fatto che pur partendo da fotografie spesso poco significative è possibile costruire immagini molto valide, utilizzando il computer solo come mezzo espressivo della propria fantasia e creatività. Quest'aspetto non va mai dimenticato: il computer non crea, esegue. Sarebbe come pensare ad un pennello in grado di disegnare da solo o ad una macchina per scrivere capace di comporre autonomamente temi e poesie.

Dal punto di vista tecnico MC può insegnare tante cose: la creatività, di contro, è un dono innato. Bravo Giancarlo!



questa quantità cromatica) mentre blu e verde sono gli stessi, come prima, del pixel adiacente. Infine se troviamo 11 i rimanenti sei pixel specificano il verde e come al solito le rimanenti quantità cromatiche sono quelle del pixel precedente. Grazie a questa tecnica HAM8 è possibile codificare oltre 256.000 colori con soli 8 bit/pixel nonostante il fatto

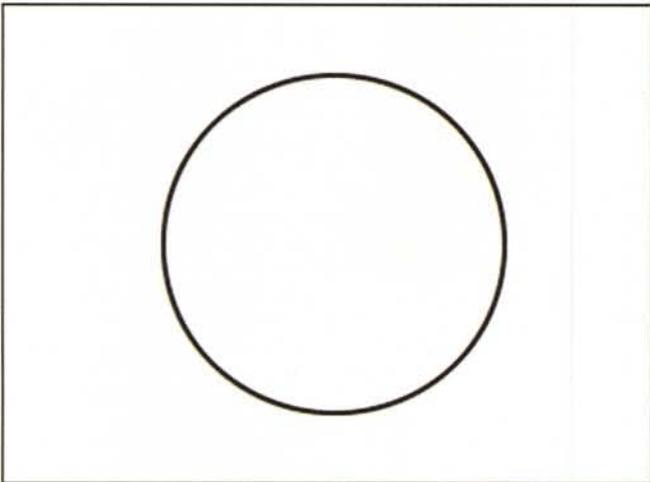
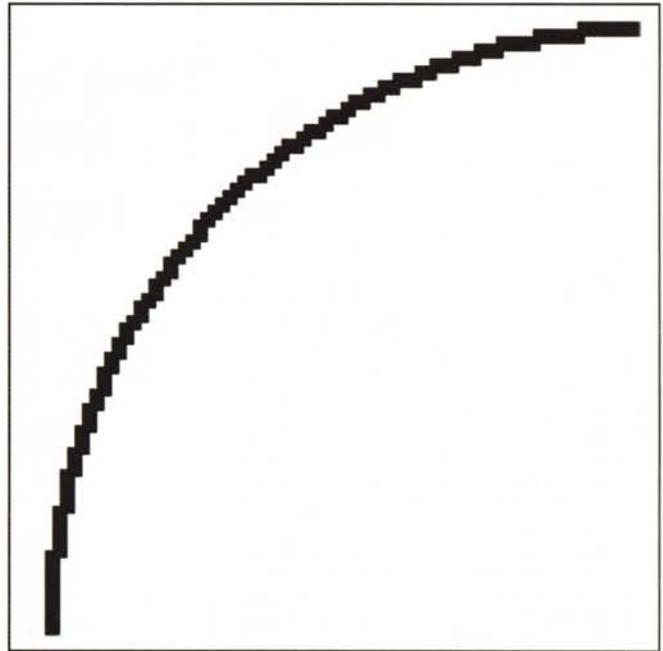
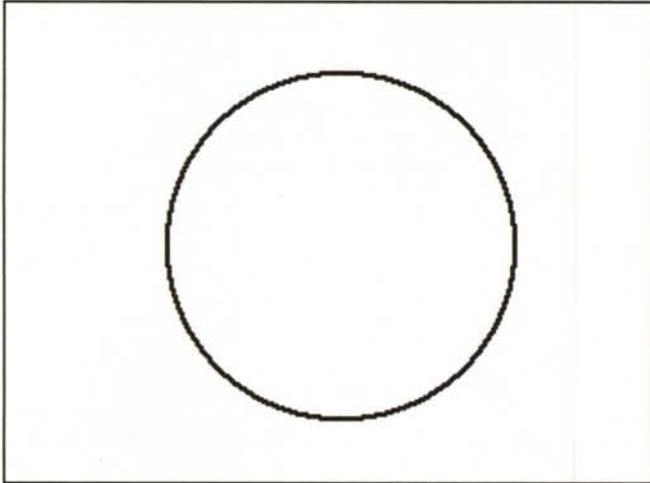
che con il metodo tradizionale ne avremmo dovuti utilizzare ben diciotto (più del doppio).

Il rovescio della medaglia (nulla è concesso gratuitamente!) sta nel fatto che per passare da un qualsiasi colore ad un altro, sempre tra gli oltre 256.000 possibili, è facile «sprecare» fino a due pixel di colore intermedio (per questo

motivo è necessario calcolare nel migliore dei modi i 64 colori di base). Se nella nostra immagine ci sono bruschi cambiamenti di colore, e non è più possibile «pescare» nella palette, tali cambiamenti saranno visualizzati solo come una (spesso antiestetica) transizione cromatica tra le due tinte larga al massimo due pixel. Quando invece si tratta

di mostrare immagini per loro natura molto sfumate, tipo incarnati, meglio se primi piani, l'effetto è molto attraente sino al punto di non notare alcunché di strano nell'immagine che stiamo guardando.

In queste quattro immagini è «svelato» il meccanismo dell'antialiasing: la risoluzione è la stessa ma la qualità apparente differisce molto.



Risoluzioni e Antialiasing

Se da una parte abbiamo appena visto che più colori utilizziamo più spazio occuperà la nostra immagine in memoria, non dobbiamo dimenticare che il dato principale riguardo la dimensione finale dipende dal numero totale di pixel utilizzati. Banalmente, come abbiamo già detto, più la nostra immagine sarà ricca di dettagli, più spazio occuperà in memoria.

Anche nel campo della risoluzione esiste un truccetto per risparmiare spazio. E non mi sto riferendo all'eventuale compressione di un'immagine una volta salvata (come avviene ad esempio, con modalità ed effetti ben diversi, con i file PICT, GIF, JPEG, ecc.) ma all'occupazione reale dell'immagine una volta caricata

in memoria. Tale tecnica è detta «antialiasing» e permette di avere una maggiore risoluzione apparente. Si basa sull'utilizzo di sfumature intermedie per simulare dettagli inesistenti. Se è vero che un'immagine vale più di mille parole, date uno sguardo ai due cerchi mostrati in questa pagina. Il secondo sembra tracciato utilizzando una risoluzione maggiore: il cerchio è più tondo, tant'è che si vede molto meno la scalettatura dovuta ai pixel. In realtà

la risoluzione utilizzata è la stessa per entrambe le immagini ed è pure piuttosto bassa. Il secondo cerchio è tracciato utilizzando il meccanismo dell'antialiasing: nei due particolari ingranditi possiamo ben notare la differenza. Alcuni punti di tonalità intermedia sono posizionati in modo tale da ammorbidire gli scalini dei pixel. L'effetto finale, pura illusione ottica, parla da sé: sim... sala... bim! =

MS