Effetti speciali nei prodotti di grafica bitmap

Ormai tutti i prodotti di grafica bitmap, dai più economici, come il famoso shareware Paint Shop Pro, ai più diffusi, come il CorelPhoto-Paint o il Micrografx Picture Publisher, ai più professionali, come l'Adobe Photo Shop, dispongono di un gran numero di Effetti Speciali. Si tratta di funzionalità che agiscono su tutta l'immagine o su una sua selezione e che producono modifiche più o meno profonde dell'immagine originale. Scopo di questo articolo è quello di curiosare un po' tra questi effetti speciali e di «ragionarci sopra» facendo anche dei semplicissimi esperimenti con il Visual Basic, in pratica dei tentativi di costruzione di effetti speciali

di Francesco Petroni

I numeri di un'immagine Bitmap

Come noto a tutti, un'immagine Bitmap (e quindi un file contenente un'immagine bitmap, ad esempio un'immagine fotografica) corrisponde ad un insieme di punti colorati, a ciascuno dei quali, in genere, al momento della visualizzazione, si fa corrispondere un pixel sul video.

L'insieme dei punti rientra, quasi sempre, in un rettangolo. Conseguentemente una delle caratteristiche fondamentali dell'immagine è la sua dimensione in punti (ad esempio 320 per 240, oppure 1.000 per 1.200, oppure ancora, se l'immagine deve corrispondere ad una videata VGA «piena», 640 per 480 punti).

Vedendo il problema da un punto di vista squisitamente numerico un'immagine bitmap corrisponde quindi ad una matrice bidimensionale di punti.

Per quanto riguarda i colori, ogni punto dell'immagine è caratterizzato da un proprio colore, che può essere memorizzato facilmente in forma numerica. Basta memorizzare i tre valori, uno per ciascuno dei colori fondamentali (Rosso, Verde e Blu, cui corrisponde il «modello RGB»).

In definitiva un'immagine può essere memorizzata in tre matrici, una per ogni colore (ad esempio R(X,Y), G(X,Y), B(X,Y)), oppure in una matrice a tre dimensioni (ad esempio C(X,Y,N), in cui N può essere 1, 2 o 3).

Ad esempio, pensando al punto 6,9 della ipotetica immagine, nel primo caso avremo:

R(6,9) = 200 quantità di rosso del punto 6,9

G(6,9) = 100 quantità di verde del punto 6,9

B(6,9) = 0 quantità di blu del punto 6,9

oppure, nel secondo caso:

P(6,9,1) = 200 quantità di rosso del punto 6,9

P(6,9,2) = 100 quantità di verde del punto 6,9

P(6,9,3) = 0 quantità di blu del punto 6,9.

Insomma si tratta in ogni caso di matrici di numeri e di numeri, che, in quanto tali, non chiedono di meglio che di essere manipolati con specifiche routine di calcolo il cui risultato sarà costituito da altri insiemi di numeri che possono a loro volta essere visualizzati producendo un'immagine differente da quella originaria.

Le routine di calcolo possono riguardare semplicemente i valori dei colori, ad esempio in un'immagine tendente al rosso si può diminuire del 10% il valore della componente di Rosso di ciascun punto, oppure si può invertire il valore numerico di ciascun colore, per ottenere una copia «negativa». Vedremo poi come.

Le manipolazioni possono anche riguardare la posizione di ciascun punto, che può essere fatta variare.

Ad esempio la semplice inversione dei valori X e Y provoca la rotazione dell'immagine.

Manipolazioni più complesse, ma sempre di questo tipo, possono essere quelle che modificano la posizione del singolo punto attraverso dei calcoli di tipo trigonometrico, quando si voglia ad esempio simulare l'effetto causato dall'avvolgimento dell'immagine sulla superficie di una sfera.

In definitiva la problematica «Effetti Speciali» può essere affrontata da due punti di vista: da quello matematico-trigonometrico, in cui si studiano gli algoritmi di manipolazione dei numeri che caratterizzano l'immagine, e da un punto di vista pratico-operativo, in cui l'utente dispone di sue immagini bitmap, magari scannerizzate e salvate in un formato bitmap «ricco», e le sottopone ad uno degli infiniti effetti speciali disponibili nei prodotti che sta utilizzando.

I formati dei file Bitmap

In Windows esiste un formato bitmap per antonomasia (il file corrispondente ha desinenza BMP) che memorizza integralmente le informazioni numeriche, esattamente nello stesso modo con il quale le stesse vengono trasferite sulla memoria video.

Il formato, che come detto si chiama BMP, è il più veloce ad essere visualizzato, ma è anche il più costoso in termini di fabbisogno di memoria. Se ne può calcolare esattamente l'occupazione moltiplicando la sua larghezza per la sua altezza e per il numero di colori.

Il formato BMP, che è facilmente producibile con il Paintbrush, è il formato usato per gli «sfondi» di Windows. Può avere 16, 256 o 16 milioni di colori (4, 8 o 24 bit per memorizzare il colore di ciascun punto).

Sono stati messi a punto decine di altri formati che comprimono il file, in modo da fargli occupare meno memo-

ria. In certi tipi di file si può decidere il fattore di compressione, che, se troppo elevato, può comportare la perdita di informazioni di dettaglio.

Gli algoritmi di compressione, in fase di scrittura, e di decompressione, in fase di lettura, rallentano le operazioni di visualizzazione.

Oltre al BMP, citato poco fa, i formati più diffusi sono il PCX, nato con il Paintbrush, il GIF, definito per le necessità di Compuserve, il JPEG, messo a punto dall'organizzazione ufficiale degli esperti in Computer Grafica. Questi due ultimi formati sono, ad esempio, utilizzabili nelle pagine WEB di Internet.

Tutti i prodotti di grafica Bitmap riconoscono tutti i principali formati, per cui non nascono mai problemi né per visualizzare né per riutilizzare le varie immagini che si trovano in giro (ad esempio nei CD).

Recentemente nei prodotti più evoluti (ad esempio CorelPhoto-Paint o Micrografx Picture Publisher) sono state inserite funzionalità che permettono di definire e manipolare «oggetti» bitmap all'interno di immagini bitmap. Questo rende i relativi formati file (rispettivamente CPT e PPX), in cui vengono memorizzati «a parte» anche gli oggetti, non totalmente compatibili con gli altri. Se si converte il formato di un'immagine realizzata con uno di questi prodotti in un altro formato si perdono le informazioni relative agli oggetti.

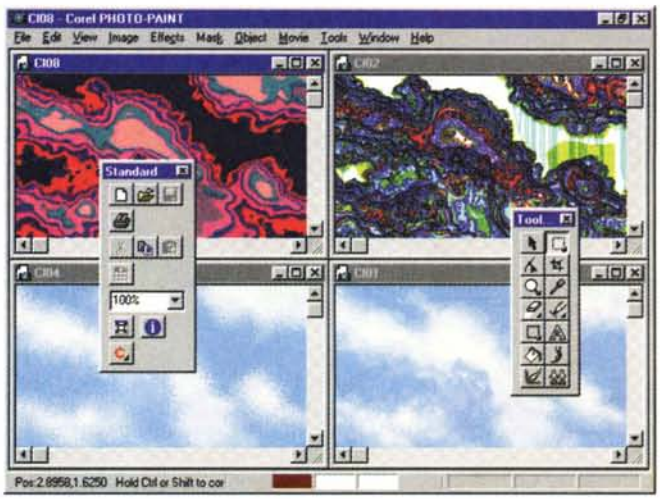


Figura 1 - Effetti speciali con Corel PHOTO-PAINT 6 per Windows 95. Con Windows 95, ancora più dotato da un punto di vista potenza grafica del suo predecessore, avrà un'ulteriore spinta la diffusione dei prodotti di grafica Bitmap (o Raster). L'immagine, come punti e colori dei punti, viene sempre memorizzata, nel file oppure nella memoria del computer, in forma numerica. Su tali numeri possono essere attivate una serie di procedure di calcolo, più o meno complesse, che modificano i numeri e quindi modificano l'aspetto della figura.

I prodotti

Nelle prime figure dell'articolo vediamo videate prese da quattro prodotti differenti. Cominciamo con CorelPHOTO-PAINT, nella versione 6 per Windows 95, e Micrografx Picture Publisher 6, anche questo per Windows 95. Vediamo, nelle prime due figure, quattro immagini ottenute applicando una serie di Effetti Speciali ad una stessa immagine iniziale, lo sfondo a «nuvolette» di Windows 95.

Nella terza figura vediamo la funzione di Preview dell'Effetto Speciale (disponibile in quasi tutti i prodotti) che con-

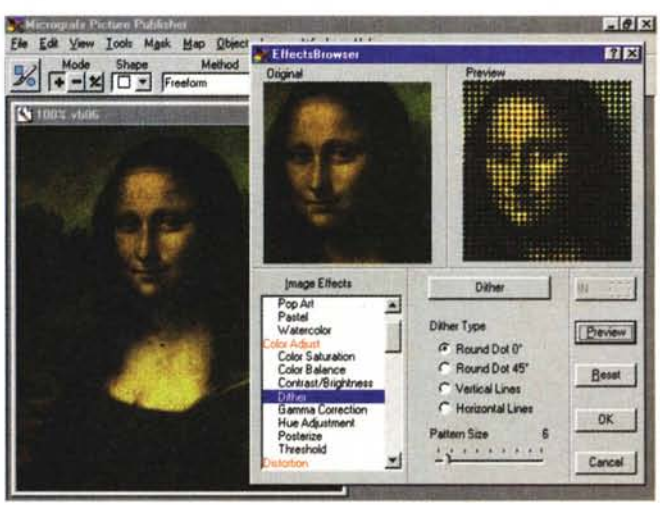


Figura 2 - Effetti speciali con Micrografx Picture Publisher 6 per Windows 95.
Una prima categorizzazione degli effetti che agiscono sulle immagini bitmap potrebbe essere quella che distingue tra procedure che agiscono solo sui colori, procedure che agiscono solo sulla posizione dei punti, e procedure che agiscono sia sui colori che sulla posizione dei punti.

Figura 3 - Micrografx Picture Publisher per Windows 95 - Preview dell'effetto speciale.
In molti casi gli Effetti Speciali sono talmente complessi che è pressoché impossibile prevederne il risultato. È quindi utilissima la funzionalità di preview che permette non tanto di scegliere il tipo di effetto quanto di definire il valore dei vari parametri in gioco.

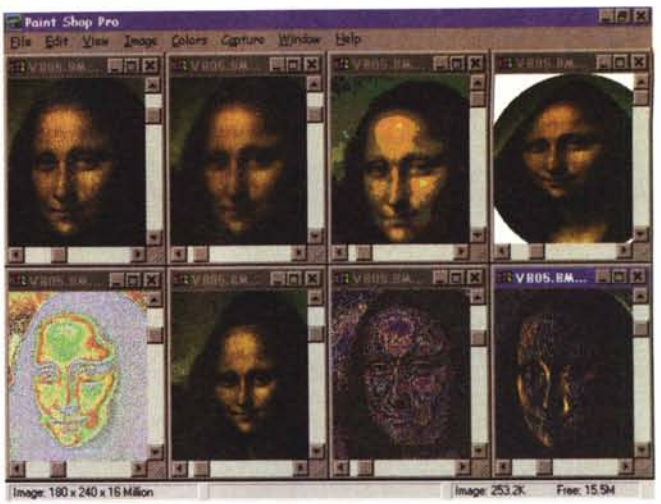


Figure 4, 5 - Paint Shop Pro 3.0 - Anche lui, nel suo piccolo...
Questo è il Paint Shop Pro, versione 3.0, prodotto di classe shareware «uscito» anche in un numero, dedicato alla Computer Grafica, di MC Software. L'ultima versione dispone di una sezione Effetti Speciali abbastanza ricca e stimolante, dotata anche di preview. Questo prodotto va benissimo per chi deve solo manipolare globalmente le immagini senza dover intervenire pesantemente per eseguire ritocchi.

sente di controllare il risultato dell'operazione prima di eseguirla in maniera definitiva.

Nella quarta e quinta figura vediamo due videate dal Paint Shop Pro, il famoso shareware, che nella recente versione 3.0, ancora a 16 bit ma funzionantissima in Windows 95, dispone di un paio di dozzine di effetti speciali, anche di tipo «geometrico» e supportati da funzioni di Preview. Interessantissima è la possibilità di creare effetti speciali personalizzati, impostando uno specifico algoritmo matematico statistico.

Chiudiamo questa minipanoramica con il classico Paintbrush per Windows 95, che però rimane un po' indietro rispetto agli altri pur essendo il prodotto di tipo bitmap più utilizzato (e spesso a sproposito) in assoluto.

Va infine ricordato che tra i prodotti del pacchetto CorelDRAW è tradizionalmente presente il CorelTRACE, che serve per vettorializzare un'immagine bitmap, individuandone le areole di colore omogeneo, che vengono trasformate in poligoni irregolari. Se invece l'immagine originale contiene linee, il risultato sarà un file vettoriale contenente linee irregolari.

Si possono ovviamente settare numerosi parametri che indicano i fattori di «arrotondamento» del processo, per quanto riguarda le areole e per quanto riguarda le linee. Lo scopo è quello di avere elementi vettoriali più regolari. È possibile settare anche il range di variazione entro il quale un colore viene giudicato omogeneo.

Nelle due figure 7 e 8 vediamo il Trace all'opera su un'immagine della Gioconda e poi l'immagine risultante trasportata in CorelDRAW, per eventuali successivi trattamenti. Qui appaiono evidenti le areole. L'effetto è quello di una Gioconda costruita con ritaglietti di carta colorata.

Mentre la vettorializzazione di un'immagine bitmap è un'operazione complessa, che non può produrre immagini vettoriali regolari, il processo inverso è quasi banale. Tutti i prodotti citati, escluso il solito Paintbrush (che però ora si chiama Paint), permettono di leggere immagini vettoriali (ad esempio in formato WMF, oppure DXF, oppure CGM, ecc.) e di convertirle e poi, eventualmente, di salvarle in bitmap.

Figura 6 - Paintbrush di Windows 95 - Pochissimi effetti... poco speciali.
Questo prodotto, che per molti utenti di Windows ha costituito l'unico «contatto» sia con la grafica di tipo bitmap sia con la grafica in genere, è rimasto al palo. Anche la sua ultima versione, per Windows 95, non dispone di alcun effetto speciale o quasi. Tra gli accessori di MS Office 95 c'è un nuovo strumento per la grafica bitmap, si chiama MS Imager.

I colori come numeri

Anche il più distratto utilizzatore di Windows si sarà accorto, o maneggiando il Pannello di Controllo o giochicchiando con il Paintbrush, che ai vari colori disponibili corrispondono dei numeri. Il sistema «internal» di Windows è quello cosiddetto additivo che è basato sui tre componenti Rosso, Verde e Blu.

Un colore quindi può essere definito impostando i tre valori delle sue tre componenti RGB.

La quantità di ciascuna componente può variare, per Windows, tra 0 e 255, per cui le combinazioni possibili sono 256 per 256 per 256, ovvero 256 alla terza, ovvero i famosi 16 milioni di colori di Windows (16.777.216, per l'esattezza).

Ad esempio se le dosi sono 0,0,0, il colore risultante dalla somma dei tre valori è il Nero, se le dosi sono 255,0,0, il colore risultante è il Rosso puro, se le dosi sono 127,127,127, il colore risultante è un Grigio intermedio, se le dosi sono 255,255,0, il colore è il Giallo puro, se le dosi sono 255,255,255, il colore è il Bianco puro.

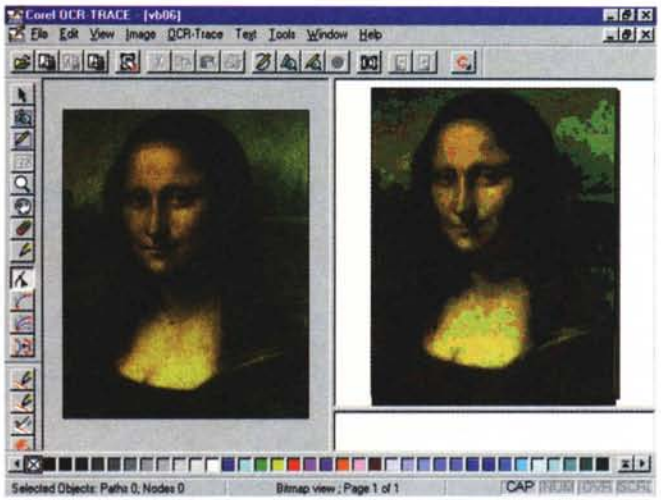
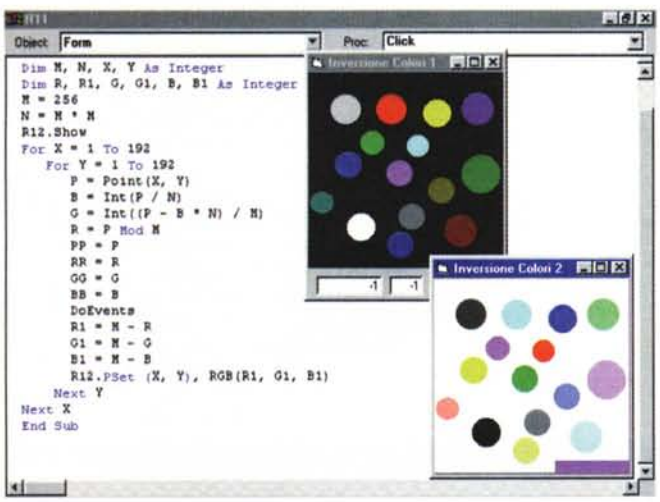


Figure 7 e 8 - CorelTRACE e CorelDRAW - Trasformazione del bitmap in un'immagine vettoriale.
Esistono numerose tecniche per tradurre un'immagine bitmap in un'immagine vettoriale. In pratica l'algoritmo individua areole di colore omogeneo e le trasforma in tanti poligoni irregolari, identificati, nel file vettoriale, con le coordinate dei nodi del loro perimetro. Nel caso in cui l'immagine originaria sia fatta di linee il risultato è costituito da una serie di linee in genere un po' sporche. Qui vediamo il CorelTRACE che trasforma l'immagine, secondo una serie di parametri che possono migliorare la resa finale, e il CorelDRAW che può utilizzare l'immagine vettorializzata.



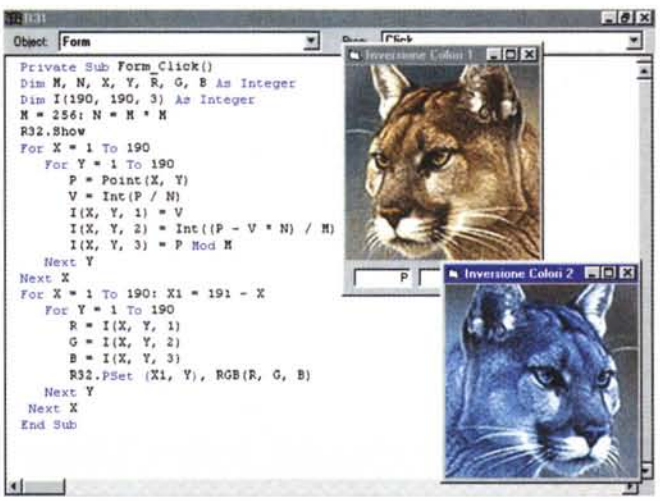


Figura 9 - Visual Basic 4.0 - Effetti Speciali su immagini Bitmap - Inversione.
Questa è la routine fondamentale che scandisce tutti i punti dell'immagine, ne legge il valore numerico (unico) del colore e lo scompone nei suoi tre componenti R, G, B. L'istruzione che «legge» il colore del punto X,Y è Point. Eseguita l'inversione dei colori (RR=256-R, ecc.) il punto viene ritracciato nella seconda finestra con l'istruzione PSet(X,Y). Per lavorare in Pixel abbiamo impostato la proprietà DrawMode della Form a 3, che significa Pixel.

Figura 10 - Visual Basic 4.0 - Effetti Speciali su immagini Bitmap - Memorizzazione in Matrice.
In questo secondo esercizio abbiamo rigirato i valori delle X (X1=191-X) per ottenere un'immagine speculare. Per potere eseguire tale operazione senza perdere le informazioni precedenti abbiamo dovuto «parcheggiare» i valori numerici, con i colori dei pixel dell'immagine, in una matrice M(X,Y,3) e poi abbiamo lavorato su tale matrice.

Potete provare qualche combinazione nella sezione «Definisci Colori Personalizzati» dell'applicazione Colori del Pannello di Controllo di Windows.

MS Visual Basic utilizza, per l'impostazione dei colori, questo stesso sistema nel senso che dispone di una funzione RGB(r,g,b) con la quale è possibile definire il colore desiderato per un determinato oggetto, passando direttamente queste tre quantità.

Visual Basic inoltre permette di leggere il colore di un punto preciso del video.

Il risultato è però un numero unico, che va quindi suddiviso nelle sue tre componenti. Vedremo poi come.

Esistono altri modelli con i quali è possibile definire un colore. Il secondo in ordine di importanza (anche questo è controllabile nella sezione Definisci Colori Personalizzati citata prima) è quello che si basa su i tre valori di Tonalità, Saturazione ed Luminosità (modello HUE). Ne esistono numerosi altri, derivati dalle tecniche tipografiche, che sono utilizzati solo nei prodotti più professionali. Non sono «visti», neanche l'HUE, da Visual Basic, per cui in questa trattazione li trascuriamo.

Come «lavora» un effetto speciale

Allora, come detto, l'immagine bitmap è completamente rappresentata da numeri, che identificano sia la posizione dei singoli punti sia il loro colore. Un Effetto Speciale esegue su tali numeri un processo di calcolo, che può essere semplicissimo, se riguarda solo la modifica dei colori (ad esempio una riduzione del 10% di rosso), oppure complessissimo se comporta la modifica della posizione dei punti.

Citiamo una serie di effetti basati su

```
' PIXEL - LATE
Private Sub Form_Click()
Dim M, N, X, Y, R, G, B As Integer
M = 256: N = M * M: R22.Show
For X = 3 To 240 Step 5
   For Y = 3 To 300 Step 5
      P = Point(X, Y)
      B = Int(P / N)
      G = Int((P - B * N) / M)
      R = P Mod M
      C = RGB(R, G, B)
      R22.Line (X - 2, Y - 2)-(X + 2, Y + 2), C, BF
   Next Y
Next X
End Sub

' SCOMPOSIZIONE ROSSO VERDE BLU
Private Sub Form_Click()
Dim M, N, X, Y, R, G, B As Integer
M = 256: N = M * M: R42.Show
For I = 1 To 190
   For L = 1 To 170
      P = Point(I, L): PP.TEXT = P
      B = Int(P / N): X = 255 - B
      G = Int((P - B * N) / M): Y = 255 - G
      R = P Mod M: Z = 255 - R
      RR = X: GG = Y: BB = Z: DoEvents
      R42.PSet (I, L), RGB(R, G, B)
      R42.PSet (195 + I, L), RGB(X, X, X)
      R42.PSet (I, 175 + L), RGB(Y, Y, Y)
      R42.PSet (195 + I, 175 + L), RGB(Z, Z, Z)
   Next L
Next I
End Sub

' GIOCONDA
Private Sub Form_Click()
Dim M, N, X, Y, R, G, B As Integer
M = 256: N = M * M: R52.Show
For X = 1 To 240
   For Y = 1 To 300: P = Point(X, Y)
      B = Int(P / N)
      G = Int((P - B * N) / M)
      R = P Mod M
      D = (300 + Y) / 600
      R52.PSet (X * D, Y), RGB(R, G, B)
   Next Y
Next X
End Sub
```

Figura 11 - Visual Basic 4.0 - Effetti Speciali su immagini Bitmap - Listati.
Qui vediamo i listati relativi ai tre programmi che corrispondono alle tre figure dalla 12 alla 14. Si tratta di variazioni sul tema rispetto al programma precedente. Le istruzioni, sempre presenti, sono quelle che servono per scorrere tutti i punti dell'immagine (due cicli For .. Next), quelle che servono per leggere le informazioni del pixel e quelle che servono per scrivere il nuovo punto nella nuova finestra.

trattamenti matematici del colore:

- Inversione dei colori di un punto P(R,G,B):
  il nuovo colore P(255-R,255-B,255-G)
- Scomposizione nei tre colori componenti di un punto P(R,G,B) e loro rappresentazione in toni di grigio:
  il primo diventa PR(R,R,R)
  il secondo diventa PG(G,G,G)
  il terzo diventa PB(B,B,B)
- Conversione in toni di grigio di un punto P(R,G,B) in cui la media sia M=(R+G+B)/3
  il nuovo colore P(M,M,M)
- Scurimento del 10% il colore del punto P(R,G,B)
  il nuovo colore P(R*1.1,G*1.1,B*1.1)

oppure calcoli più raffinati basati non sulla moltiplicazione ma sulla proporzione.

- Omogenizzazione dei colori eseguendo dei calcoli statistici sui colori dei vari punti in modo da addolcire le discontinuità ed eliminare i punti sporchi.

Se invece ipotizziamo anche la modifica della posizione del punto P(X,Y) dell'immagine (le cui dimensioni siano L,H) potremo eseguire i seguenti trattamenti:

- Rotazione planare di 90 gradi impostando il punto P(Y,X)
- Incorniciatura dell'immagine rettangolare in un'area circolare
- Effetto speculare lungo la verticale impostando il punto P(L-X,Y)
- Effetto di rotazione di A gradi impostando P(X*cos(A),Y*sin(A))

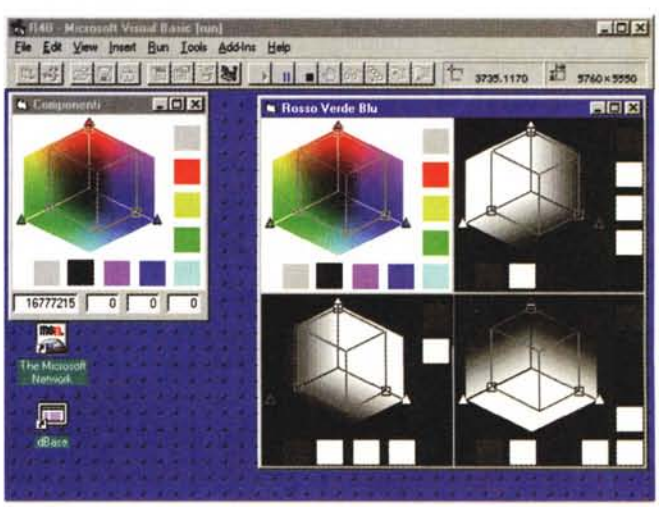
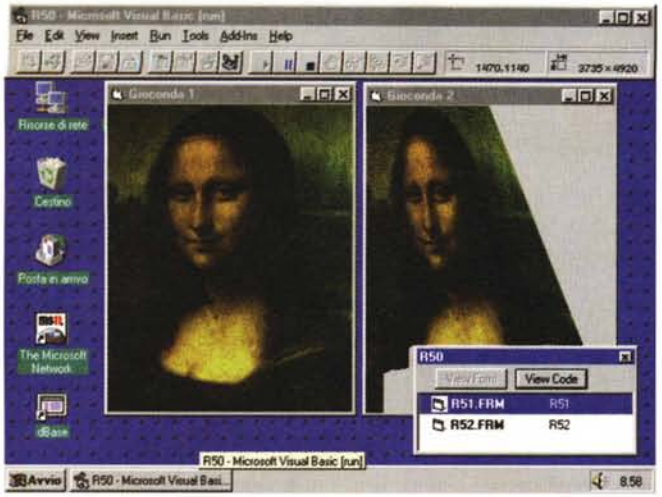


Figura 12 - Visual Basic 4.0 - Effetti Speciali su immagini Bitmap - Pixel-late.
Per ottenere l'effetto «mosaico» leggiamo solo alcuni punti dell'immagine (ne leggiamo uno su cinque in orizzontale e uno su cinque in verticale e quindi, in totale, uno su venticinque) e li riproduciamo in un quadratino di 5 per 5 pixel. È chiaro che tale valore 5 si può facilmente parametrizzare.

Figura 13 - Visual Basic 4.0 - Scomposizione nei tre colori fondamentali.
Abbiamo scomposto questa immagine, che rappresenta un modello classico di rappresentazione dei colori, nelle sue tre componenti cromatiche. In pratica il livello di grigio di ognuna delle tre componenti rappresenta il valore di quel componente nell'immagine completa. Risulta chiaro questo concetto vedendo i rettangolini con i colori fondamentali. Ad esempio al rosso puro corrisponde il nero nella rappresentazione dei Rossi e il bianco in quelle dei Verdi e dei Blu. Da un punto di vista numerico sarebbe il colore RGB(255,0,0).



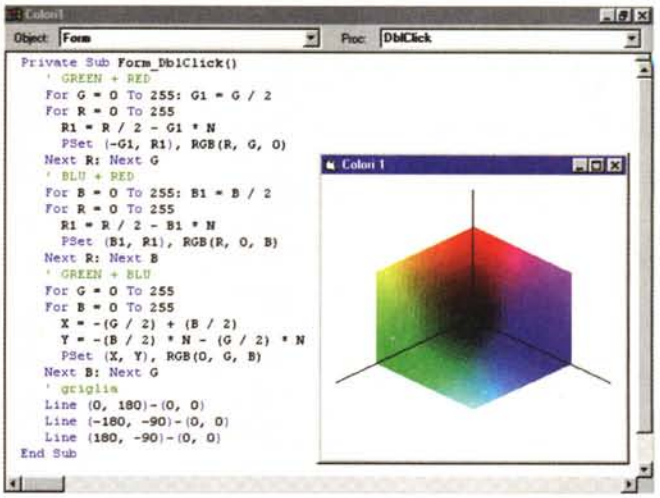


Figura 14 - Visual Basic 4.0 - Deformazione dell'immagine.
In questo caso abbiamo imposto una semplice deformazione geometrica alla nostra immagine. Il valore del punto X lo abbiamo dimezzato. È chiaro che sono possibili infinite varianti che possono far assumere qualsiasi forma all'immagine risultante.

Figura 15 - Visual Basic 4.0 - Costruzione di un modello per i colori.
Questo modo di rappresentare i colori secondo il modello RGB è comprensibilissimo. Ogni asse rappresenta un componente, il cui valore varia da 0 fino a 255. Quindi abbiamo gli assi R, G e B. Il piano individuato da R e G mostra tutti i colori ottenuti miscelando differenti quantità di tali due componenti. Lo stesso dicasi per le coppie R e B, G e B. Questo modello non rappresenta affatto i colori caratterizzati da tre componenti. Se si volessero tracciare anche questi occuperebbero lo spazio antistante i tre piani, coprendoli e coprendosi l'uno con l'altro.

Altri effetti possono essere derivati dalla realtà oppure da altre tecnologie:

- Effetto mosaico ottenuto «mediando» un'areola di pixel e convertendola in un'areola di uguali dimensioni, ma del colore medio
- Effetto illuminazione in cui viene simulata la presenza di una lampada che illumina in maniera non uniforme (con un cono di luce) l'immagine
- Effetto trasparenza, in cui l'immagine si «somma» ad altre immagini per simulare ad esempio una carta che fa da sfondo, oppure un vetro che la copre, ecc.
- Effetti ottici, ad esempio la sfocatura dei punti di discontinuità dell'immagine
- Effetti fotografici, come la solarizzazione che simula la sovraesposizione della stampa
- Effetti pittorici, quando l'areola di pixel subisce un processo simile a quello tipico della pennellata di un pittore, che poggia il pennello ed esegue, in una direzione a scelta, la pennellata
- Effetti scultorei, quando viene evidenziato, con un effetto rilievo, la discontinuità di colore.

Ipotizziamo poi altri effetti basati su calcoli trigonometrici per creare effetti 3D:

- Avvolgimento dell'immagine attorno ad un solido spaziale, ad esempio un cilindro, una sfera, un cono,
- Vista a volo d'uccello, con la quale si costruiscono viste prospettiche dell'immagine piana vista da un punto obliquo rispetto alla verticale del suo punto centrale.

Molti di questi effetti sono già presenti nei vari prodotti, anche in quelli citati. Ognuno di questi può essere comunque realizzato «in casa» con un'apposita routine matematica che manipola di dati numerici. Gli Effetti Speciali hanno nomi che cercano di far capire il risultato cui tendono. Poiché si tratta in genere di applicazioni non tradotte i nomi sono in inglese. Ad esempio: Blur, macchia, Pinch, pizzico, Punch, punzonatura, Skew, obliquo, Edge, orlo, Sharpen, aguzzare, Emboss, scolpire, ecc.

Come e perché affrontiamo gli Effetti Speciali con MS Visual Basic

Usiamo il Visual Basic della Microsoft perché è un prodotto molto diffuso, è facilissimo da utilizzare ed inoltre dispone di alcune istruzioni che servono al nostro scopo.

La controindicazione consiste nel fatto che un'applicazione Visual Basic è molto più lenta di una sviluppata con un linguaggio più vicino alla macchina, come il Visual C. Le esecuzioni degli Effetti Speciali che realizzeremo saranno ovviamente molto lente anche nei confronti degli identici effetti presenti nei prodotti grafici.

Facciamo una serie di considerazioni di tipo generale, comuni ai vari programmi che presentiamo:

- Usiamo il Visual Basic 4.0, ma gli stessi esperimenti proposti si possono realizzare con qualsiasi versione precedente
- Abbiamo preparato una serie di immagini campione, tra cui una Bitmap con la Gioconda delle dimensioni di circa 240 per 300 pixel a 256 colori
- In Visual Basic è possibile definire per le Form la proprietà Picture, e quindi caricare una bitmap direttamente come «sfondo» della Form
- La Form dispone della proprietà DrawMode, che, se posta uguale a 3, ci fa lavorare sulla Form direttamente in pixel
- Sulla Form agisce il Metodo Scale che ne specifica un sistema di riferimento utente.

In pratica nei nostri programmi tali impostazioni iniziali verranno legate all'evento Form_Load, ad esempio:

```
Private Sub Form_Load()
        Width = 4600: Height = 4600
        ScaleMode = 3
        Scale (-200, 200)-(200, -200)
        BackColor = RGB(255, 255, 255)
End Sub
```

I valori di queste proprietà andranno SEMPRE adattati alle immagini che si vogliono trattare.

Le istruzioni principali che utilizzeremo sono due:

```
C1 = POINT(X,Y)
```

si tratta di una funzione che restituisce un valore numerico unico indicante il colore del punto X,Y.

```
PSET(X,Y),C2
```

Questa è un'istruzione che colora con i colore C2, il punto X,Y della Form. Il colore C2 può essere specificato con la funzione RGB(r,g,b) alla quale si debbono passare i tre valori numerici delle tre componenti cromatiche (numeri che vanno da 0 a 255).

Il valore C1, che è uno solo, corrisponde alla seguente formula matematica:

```
C1=B * 65.536 + G * 256 + R
```

Ad esempio se il colore ha le tre componenti 64, 128 e 255 il valore di C1 è 4.227.327. Questo valore può es-

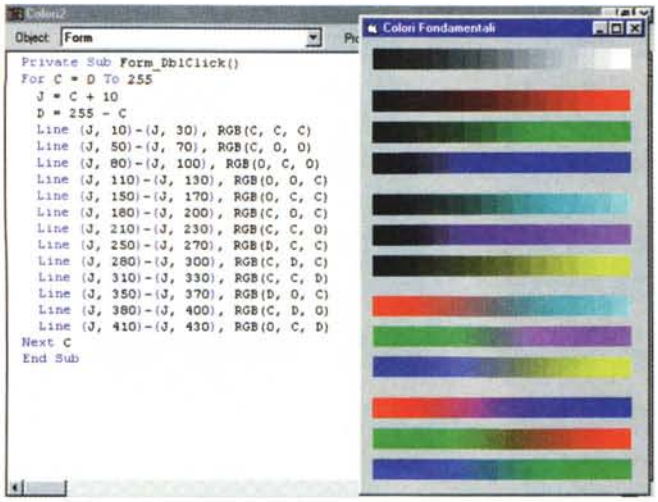


Figura 16 - Visual Basic 4.0 - Giocare con i colori.
Semplici routinette per creare effetti sfumatura tra due colori fondamentali. Anche in questo caso abbiamo approfittato del fatto che il range dei valori numerici che rappresentano i colori (da 0 a 255) è compatibile con il range dei pixel del video sul quale vogliamo vedere tali colori.

Figura 17 - Visual Basic 4.0 - La Gioconda sulla Sfera - Risultato.
Qui utilizziamo formule trigonometriche. Supponiamo in pratica di avvolgere attorno ad una sfera, a da partire da un punto che è in perfetta verticale rispetto all'osservatore, la nostra bitmap con la Gioconda. L'immagine risulta tanto più deformata quanto più ci si allonana da tale punto. Sarebbe stato più corretto, ma un po' più complesso dal punto di vista algoritmico, far corrispondere il punto verticale, citato prima, con il centro della figura.

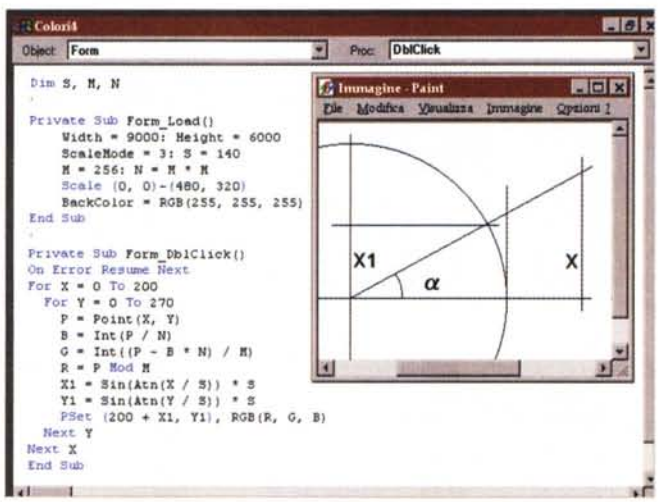


Figura 18 - Visual Basic 4.0 - La Gioconda sulla Sfera - Listato e schema geometrico.
Vediamo i tre pezzi del programma e vediamo, in primo piano, lo schema geometrico su cui abbiamo basato le formule di conversione. In pratica abbiamo dovuto trasformare ogni valore X (e quindi Y) dell'immagine originale in un valore X1 della proiezione della sua posizione riportata sulla sfera. Purtroppo Visual Basic non dispone della funzione Arcotangente (X,Y) che avrebbe consentito di lavorare facilmente anche con angoli negativi. Questo è il motivo per il quale abbiamo utilizzato un solo quadrante, quello con gli angoli positivi.

sere ottenuto solo con quella combinazione di valori dei componenti.

L'operazione contraria, parliamo della scomposizione del valore C1 nei suoi tre componenti, si può eseguire con una serie di tre operazioni:

B è uguale alla parte intera di C1 diviso per 65.536 (256 per 256),

G è uguale alla parte intera di C1 meno R * 65.536 (già attribuito al rosso) diviso per 256,

R è uguale al resto della divisione di C1 per 256 (quello che avanza dopo aver definita la quota del Rosso e quella del Verde).

Le routine relative le vedete nel listato del programma in figura 9, programma nel quale i vari colori vengono invertiti ottenendo un «negativo» dell'immagine iniziale.

Poiché l'immagine risiede direttamente sulla Form e poiché dalla Form, con l'istruzione POINT(X,Y), possono essere letti i singoli pixel, non è necessario predisporre una matrice con i valori del numero del colore.

I primi cinque esempi, che sono ben documentati dai listati e dalle didascalie (figure dalla 10 alla 15), producono semplici manipolazioni delle figure di origine.

Abbiamo sempre usato due Form per poter vedere contemporaneamente le immagini prima e dopo... la cura.

Nelle fasi iniziali di test, specialmente nel test dei calcoli numerici, abbiamo piazzato sulla Form anche delle Label o delle Text Box nelle quali controllare i valori via via calcolati.

Lo scopo dei successivi due esercizi, altrettanto semplici, è quello di giocare con i colori. Nel primo esercizio ricostruiramo il modello RGB (è quello in figura 15), nel secondo creiamo delle semplici sfumature tra due colori (figura 16).

Nell'ultimo esercizo sottoponiamo i numeri che individuano la posizione dei vari punti dell'immagine ad una routine di calcolo trigonometrico che li trasforma nelle coordinate delle loro proiezioni su una superficie sferica. Usiamo solo una «fetta» della sfera per non complicare troppo il programma (anche perché il Visual Basic non dispone della funzione ATN2). Nella figura 17 vediamo la Gioconda, prima e dopo la cura, nella successiva figura 18 vediamo il listato del programma ed uno schemetto grafico della trasformazione trigonometrica realizzata.

Conclusioni

Abbiamo voluto fornirvi una serie spunti dai quali partire per ulteriori e più complessi esperimenti.

I listati sono da considerare quindi del tutto indicativi. Se li utilizzate dovete assolutamente adattare il range dei calcoli dei due cicli For .. Next alle dimensioni delle vostre immagini. Il range dei valori numerici dei colori invece va sempre da 0 a 255. Se per sbaglio indicate, o il programma indica, altri valori si genera un errore.

Va infine detto che l'istruzione PSET, in realtà, legge il Pixel sul video, per cui in fase di lettura del valore del colore dell'immagine occorre che questa non sia coperta da altre cose, ad esempio da altre finestre.

MC

Internet, BBS, Videotel, FAX, Voce, Teleassistenza, *ISDN*....

OMOLOGAZIONE CE EUROPEA

OMOLOGAZIONE ISPT IN CORSO

Linea modem Elite V.34-ISDN, Fax, Voce

L'interfaccia telematica globale: il più potente modem sul mercato mondiale. Interfaccia telefonica per linee commutate o accessi base ISDN. Velocità di 28.800 Bps (V.34) in modalità analogica e 64.000/128.000 (V.110 V.120 X.75) Bps in modalità digitale. Terminal adaptor analogico integrato per sfruttare la linea ISDN con un normale telefono. Ricezione fax anche a computer spento, porta dati parallella e seriale, funzioni voce per realizzare segreterie telefoniche, voice box, servizi di fax-back. Protocollo per telefoni cellulari ZyXCELL a 14.400 Bps. Stampa diretta dei fax sulla stampante collegata al modem.

VideoOnLine
Abbonamenti a Internet in Omaggio

ZyXEL è la soluzione

yes It runs with NetWare

Designed for Microsoft Windows 95

ISO9001

Linea modem U1496 V.32bis 19200 Bps, Fax, Voce

Importa, distribuisce e assiste:

SIDIN
SOCIETÀ ITALIANA DI INFORMATICA s.r.l.

Via Canova, 25 - 10126 Torino
Tel. 011/6633863
Fax 011/3100493

Internet: www.inrete.it/sidin/sidin.html

Per maggiori informazioni compilare ed inviare via fax o per posta a SIDIN srl:

NOME e COGNOME

SOCIETÀ

VIA

CAP CITTÀ

TEL. FAX

☐ RIVENDITORE ☐ AZIENDA ☐ PRIVATO

MC