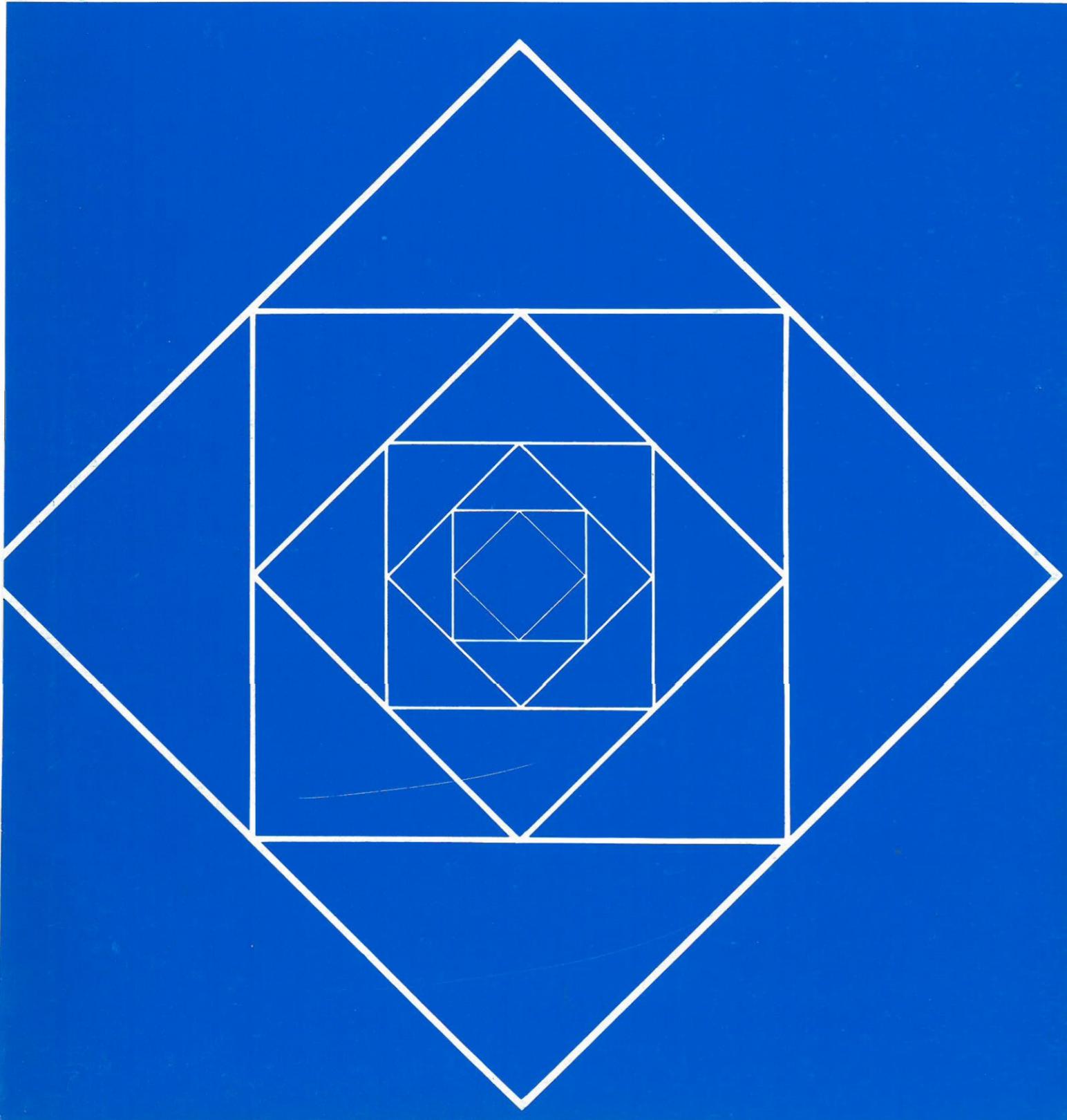


## I formati grafici e l'interscambio di immagini

IBM Internal Use Only

G. Chiozzi



Per solo uso interno IBM

# **I formati grafici e l'interscambio di immagini**

di

*G. Chiozzi*

RIP2RGB	36
RIP2VGA	37
RIPCLIP	37
RGBCONV	38
RGBCONV2	39
SHI6090	40
TIFF2PB	40
TORIP	41
VGA2PMBM	42
<b>Appendice A. Mappa delle conversioni</b>	<b>43</b>
<b>Appendice B. I formati più diffusi</b>	<b>45</b>
<b>Appendice C. Files utilizzati</b>	<b>47</b>
<b>Appendice D. I programmi sul dischetto allegato</b>	<b>49</b>
<b>Appendice E. PBRUSHUT: i programmi del package</b>	<b>51</b>
<b>Appendice F. Utili prodotti software e TOOLS</b>	<b>53</b>
<b>Bibliografia</b>	<b>55</b>

## Figure

1. Il ciclo di vita di una immagine . . . . .	3
2. Immagini raster e vector a confronto . . . . .	4
3. I sistemi di coordinate RGB e hsv . . . . .	6
4. Lo schema utilizzato per le conversioni. . . . .	10
5. I parametri di una immagine raster . . . . .	11



## Introduzione

La nascita di nuovi strumenti per l'acquisizione e l'elaborazione delle immagini è accompagnata dallo svilupparsi di sempre diversi formati grafici per la memorizzazione delle immagini stesse.

Chi si occupa di applicazioni di computer graphics e sistemi multimediali deve ogni giorno affrontare la necessità di utilizzare questi diversi formati, con la non remota possibilità di incontrarne dei nuovi che non sono compatibili con i programmi che utilizza.

È inoltre impensabile credere di riuscire ad eliminare questo problema con una scelta accurata *a priori* degli strumenti di lavoro; infatti:

- la necessità di utilizzare pacchetti diversi e specializzati per ogni specifica parte del lavoro da svolgere (acquisizione, editing, authoring,...),
- la diversità delle fonti da cui provengono le immagini da elaborare,
- l'opportunità e la necessità di adottare i pacchetti applicativi più moderni a disposizione sul mercato

costringono ad analizzare il problema a posteriori.

Occorre inoltre tenere presente che la diffusione della "filosofia" del Personal Computer e delle applicazioni desk-top (desk-top publishing, desk-top video,...) pone di fronte a queste problematiche utilizzatori che in generale non sono né esperti di computer graphics, né programmatori.

Questa guida si propone tre diversi obiettivi:

1. Introdurre gli utilizzatori di applicazioni grafiche alla logica sottostante le scelte effettuate dagli sviluppatori dei diversi formati grafici per comprendere le esigenze che ne stanno alla base.
2. Insegnare ad utilizzare un insieme di programmi di utilità per effettuare le conversioni fra i formati attualmente più diffusi, senza che per questo siano richieste particolari competenze.
3. Fornire gli strumenti di base a persone con limitate esperienze di programmazione per sviluppare procedure in grado di risolvere specifiche esigenze.

Nei capitoli seguenti verrà pertanto fornita una descrizione introduttiva alle problematiche relative alla generazione e all'utilizzo di immagini digitalizzate.

Le informazioni di carattere più strettamente tecnico, utili a coloro che vogliono realizzare le proprie routines di conversione, sono racchiuse in "note tecniche" e chi non sia interessato può evitare di leggerle.

Il capitolo successivo descrive dettagliatamente alcuni esempi di procedure di conversione che richiedono l'esecuzione di passi successivi; lo scopo è quello di mostrare la metodologia da seguire per sviluppare proprie procedure mediante l'utilizzo di programmi già pronti, componendoli in una catena di conversioni successive; queste possono poi venire raccolte in files di comando DOS (di tipo .BAT) nel caso di un utilizzo frequente.

Segue poi un manuale di riferimento relativo a numerose routines di conversione, provenienti da diverse fonti (in particolare dal package Pc Paintbrush Utilities - PBRUSHUT - disponibile nei PCTOOLS<sup>1</sup>).

Molte delle routines descritte sono state sviluppate presso i Centri Ricerca IBM di Milano e Roma: alcune di queste sono inserite nel dischetto allegato (per informazioni sulla disponibilità del dischetto rivolgersi

all'autore), anche in formato sorgente per essere utilizzate come esempio nella scrittura delle proprie procedure di conversione e per comprendere più facilmente le caratteristiche dei formati descritti.

Seguono poi numerose appendici da utilizzare come rapidi schemi per individuare le routines necessarie per effettuare una particolare conversione e le fonti presso le quali reperirle (alcune sono anche fornite come "tools" di prodotti presenti sul mercato, come ad esempio PictureTaker, il programma fornito assieme allo StoryBoard Live IBM che permette di catturare schermate di applicazioni per utilizzarle in proprie presentazioni).

---

<sup>1</sup> Il sistema HONE mette a disposizione degli utenti diversi repository internazionali di programmi di pubblico dominio a carattere "IBM Internal Use Only", detti genericamente TOOLS. Fra questi si possono distinguere:

**AIXTOOLS** sul quale reperire applicazioni per il sistema operativo AIX su Personal Computer, RISC/6000 e IBM 6150;

**OS2TOOLS** contenente routines per il sistema operativo OS/2 su Personal Computer;

**PCTOOLS** contenente programmi per Personal Computer in ambiente DOS;

**VMTOOLS** contenente programmi in ambiente VM/CMS;

**TXTOOLS** contenente programmi per la manipolazione di testi.

Nello stesso ambiente sono anche disponibili numerose conferenze ricche di informazioni relative ai problemi di conversione di immagini e sulle quali si possono porre le proprie domande.

## I formati grafici e l'interscambio di immagini

### I files grafici

#### Il ciclo di vita di una immagine

Il ciclo di vita di una immagine computerizzata attraversa essenzialmente tre fasi:

- La generazione** Questa è la fase in cui una immagine viene creata in forma digitale, ad esempio mediante l'uso di un editor grafico (come il Picture Maker di StoryBoard Live), per acquisizione di una fotografia mediante scanner o per cattura di una schermata del Personal Computer.
- L'elaborazione** Nella maggior parte dei casi prima di poter essere utilizzata, l'immagine acquisita deve essere elaborata, ritagliandone alcune parti, aggiungendo informazioni, cambiando il numero di colori o le dimensioni o in mille altri modi.
- L'utilizzazione** Alla fine l'immagine può essere inserita in una presentazione grafica (ad esempio realizzata con l'IBM Audio Visual Connection - AVC - o con StoryBoard Live) o in un testo scritto per la realizzazione di un articolo.

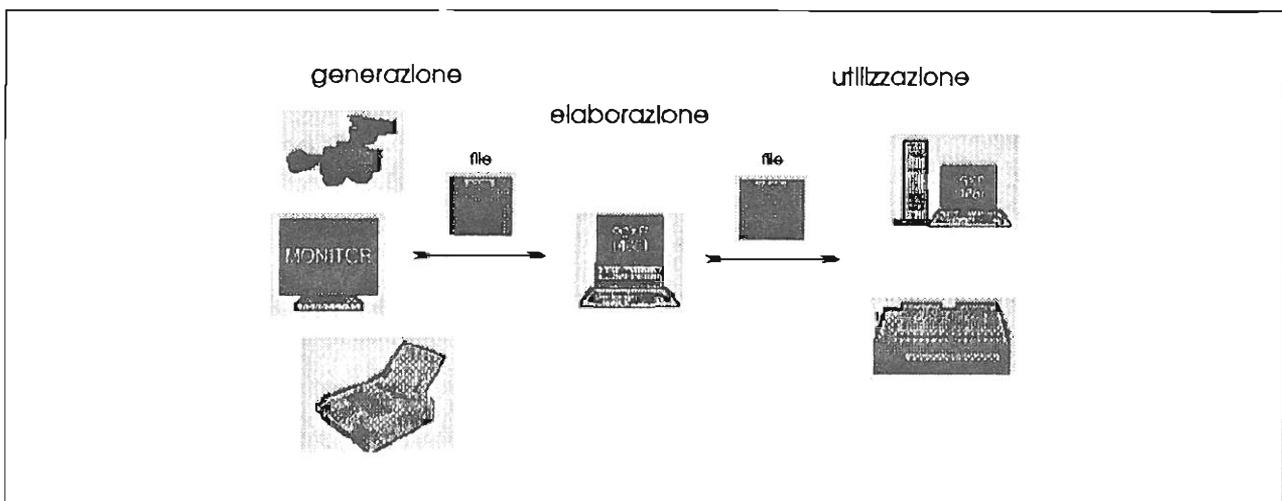


Figura 1. Il ciclo di vita di una immagine

In genere queste fasi sono realizzate utilizzando pacchetti diversi, specifici anche in funzione dei dispositivi hardware a disposizione: ad esempio, un software di acquisizione per una telecamera a colori sarà in generale ben diverso da quello per uno scanner in bianco e nero, magari di un diverso produttore. In ogni caso, comunque, lo scambio di "immagini digitali" fra le diverse applicazioni avverrà attraverso files archiviati sui dischi dei computers utilizzati ed eventualmente spediti in rete locale. La possibilità di inserire in questa catena diversi prodotti dipende quindi dalla loro capacità di interfacciarsi gli uni con gli altri, cioè di avere formati per i files in output compatibili con quelli in input degli elementi successivi del processo e viceversa.

Risulta inoltre subito evidente l'importanza della fase di archiviazione delle immagini, che deve tenere conto dei limiti di capacità dei sistemi di memorizzazione digitale, in forte contrasto (come vedremo) con l'alta densità di informazione di cui una immagine necessita per essere rappresentata. Lo stesso problema si pone nel caso di immagini trasmesse in rete, dove i tempi di risposta del sistema sono strettamente legati alla quantità di dati da trasferire e ne costituiscono, in generale, il punto critico.

Un secondo importante problema, che nasce dall'utilizzo di architetture distribuite, nelle quali convivono dispositivi hardware profondamente diversi e numerosi programmi applicativi, deriva dalla necessità di archi-

viare le immagini in formato indipendente dal dispositivo con cui sono state generate, cosicché sia possibile la loro utilizzazione su tutti i diversi sistemi di visualizzazione.

## La rappresentazione digitale delle immagini

Si distinguono due tecniche diverse per rappresentare immagini in forma digitale:

**Raster** Questo metodo è il più diffuso per le immagini di tipo fotografico, e consiste nel suddividerle in pixels (picture elements) mediante sovrapposizione di una griglia logica di dimensioni prefissate. In questo modo si ottiene una matrice di elementi che, rappresentati uno ad uno, permettono di ricostruire l'immagine. È chiaro che la fedeltà della rappresentazione dipende in maniera critica dalle dimensioni della griglia e dall'informazione associata a ciascun pixel (questi aspetti verranno meglio specificati più avanti). L'occupazione di memoria risulta molto elevata.

**Vector** In questo caso, le immagini (generalmente disegni di tipo tecnico, realizzati con programmi CAD - Computer Aided Design) sono rappresentate mediante una descrizione dei loro elementi costitutivi - linee, testo, cerchi,...- utilizzando un opportuno linguaggio. Un tipico esempio di questo tipo è il formato detto HP-GL utilizzato per i plotters proprio perché questi "disegnano" l'immagine elemento per elemento con una penna.

La rappresentazione è in genere molto compatta, ma non permette di rappresentare immagini fotografiche. Per questo motivo i linguaggi più evoluti per le rappresentazioni di immagini in formato vettoriale comprendono fra i possibili elementi costitutivi (e a fianco di linee, testo, curve, ecc.) immagini di tipo raster: è questo il caso dell'ormai diffusissimo linguaggio PostScript® utilizzato per la stampa ad alta qualità con le stampanti laser o per la gestione di sofisticate schede video (Video-PostScript).

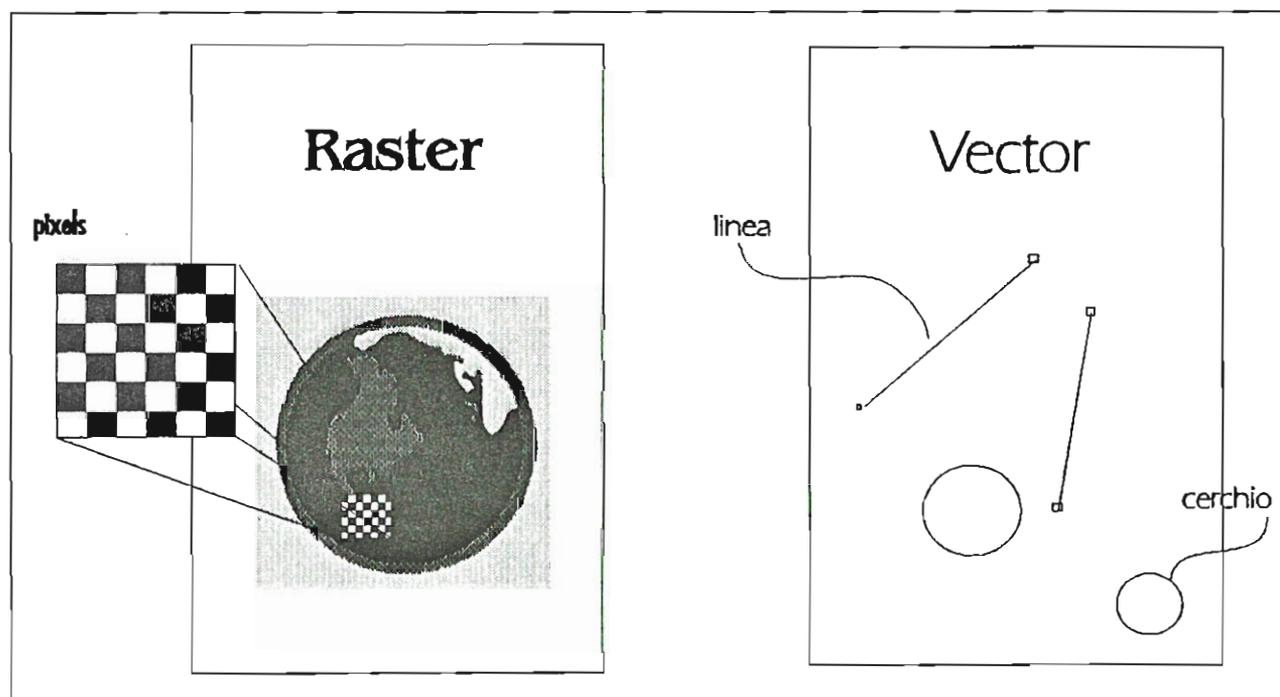


Figura 2. Immagini raster e vector a confronto

Non ci occuperemo dei formati vector, ma solo di quelli raster. Questi ultimi, come già detto, rappresentano le immagini pixel per pixel individuandone il colore: questo risultato tuttavia può essere ottenuto in diversi modi. Si possono individuare tre modelli principali (descritti nei paragrafi seguenti): RGB, a *look up table*, *memory dump*.

## I modelli RGB

In questo caso, per ogni pixel viene indicato il suo colore. Generalmente viene utilizzata per questo scopo la rappresentazione RGB (red, green, blue) mediante la quale ogni colore è rappresentato *dalle quantità dei colori di base rosso, verde e blu che lo compongono: quasi tutti i colori possono essere ottenuti, sotto una ampia gamma di condizioni di osservazione, miscelando opportune quantità di tre colori primari opportunamente scelti.* Il rosso, il verde e il blu costituiscono appunto una buona scelta per i colori primari, ma non l'unica possibile; tuttavia questa terna è una delle più usate e corrisponde al modo con cui sono ottenuti i colori sui monitors, cioè per sintesi additiva (sovrapposizione) della luce emessa da tre fosfori rispettivamente rosso, verde e blu: l'elettronica del monitor controlla, per ciascun pixel, la quantità di luce emessa per le tre bande, dalla cui sovrapposizione il nostro occhio riceve l'impressione di vedere un particolare colore.

Le stampanti a colori, invece, utilizzano generalmente la base giallo, magenta e blu, dalla quale si ottengono i colori per sintesi sottrattiva; questo significa che un colore è ottenuto sovrapponendo opportuni strati dei colori primari. In questo modo lo spazio dei colori ottenibili è più limitato.

Un altro metodo di rappresentazione molto utilizzato, perché legato ad aspetti intuitivi dei colori, è la cosiddetta base hsv (hue, saturation, value): la hue corrisponde infatti al concetto di "colore dominante", la saturation all'effetto del controllo "colore" nei televisori e il value a quello del controllo di "intensità".

Nota tecnica 1 - Le basi RGB e hsv:

Si può formalizzare quanto detto sopra rappresentando i colori come uno *spazio vettoriale a tre dimensioni*. La scelta della rappresentazione RGB corrisponde a individuare in questo spazio una base ortogonale i cui versori siano appunto R, G e B (vedi Figura 3 a pagina 6).

In questo spazio la scala dei grigi corrisponde ai vettori con uguali quantità delle tre componenti.

Poiché i tre primari costituiscono una base in questo spazio, ogni colore può essere ottenuto come loro combinazione lineare:

$$qQ = rR + gG + bB$$

tuttavia per alcuni colori questo risultato è ottenibile solo utilizzando coefficienti (cioè quantità) negativi per alcune componenti, fatto ovviamente non realizzabile in pratica. Questo fenomeno rende conto dell'affermazione secondo la quale *quasi tutti i colori sono ottenibili utilizzando una opportuna miscela dei primari*.

Scegliere una diversa rappresentazione corrisponde ad effettuare un cambio di base e la scelta di una terna di primari è limitata a vettori linearmente indipendenti (colori cioè che non possano essere ottenuti uno dalla miscela degli altri due).

Date due terne di primari esiste pertanto sempre una matrice di trasformazione che porta da un sistema di coordinate all'altro.

Questo modello tridimensionale per la rappresentazione dei colori deriva dall'osservazione sperimentale che nell'occhio umano sono presenti tre *recettori* diversi, assimilabili a tre sensori lineari per opportuni colori primari.

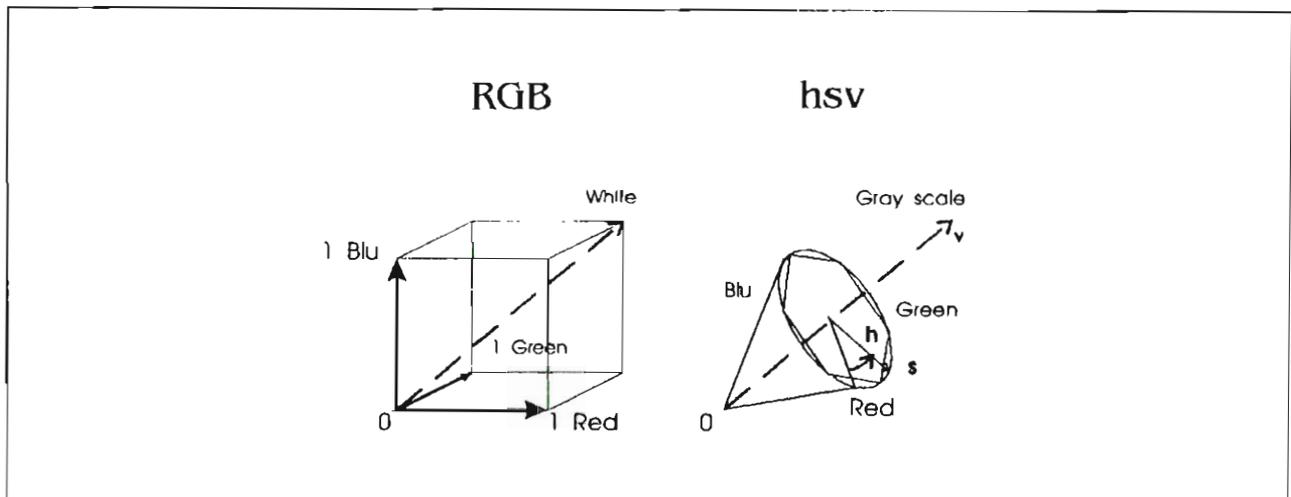


Figura 3. I sistemi di coordinate RGB e hsv

La scelta della rappresentazione hsv corrisponde invece a un cambiamento di rappresentazione più radicale: si passa infatti da un sistema in coordinate cartesiane ortogonali a uno in coordinate cilindriche (vedi Figura 3).

L'asse della rappresentazione (la coordinata  $v$ ) è la retta della scala dei grigi (con intervallo da 0 a 1), mentre la distanza dall'asse è la coordinata  $s$  (sempre con intervallo fra 0 e 1) e l'angolo di rotazione (a partire dal colore rosso puro) è la coordinata  $h$ .

Utilizzando questo sistema di coordinate si possono facilmente ottenere interessanti effetti cromatici: ad esempio, per ottenere una immagine in bianco e nero a partire da una a colori, basta mettere a 0 per ogni pixel il coefficiente per  $s$ .

Anche per questi sistemi di riferimento esistono semplici leggi di trasformazione da e verso quelli ortogonali cartesiani.

Limitiamoci comunque per semplicità a considerare il caso del modello RGB. Il numero di colori diversi che si possono rappresentare dipende dal numero di bits che si hanno a disposizione per ognuna delle tre bande, tenendo presente che il nero si ottiene ponendo a 0 tutti e tre i valori e il bianco ponendoli al massimo.

Il valore assunto da una delle tre bande rappresenta infatti l'intensità della luce emessa dal corrispondente fosforo del monitor, dove 0 significa fosforo spento e  $2^n$  (2 elevato a  $n$ ) emissione massima (potenza al 100%); pertanto il range di potenza del fosforo risulta suddiviso in  $2^n$  possibili intervalli.

Se, ad esempio, si hanno a disposizione 16 bits in tutto (come avviene per le immagini acquisite mediante la scheda VCA), se ne possono assegnare 5 alla banda rossa (red), 6 a quella verde (green) e ancora 5 a quella blu (l'occhio umano è più sensibile alle variazioni di colore nel verde e quindi normalmente si assegnano più bits a questa banda). In questo caso si hanno a disposizione  $2^5 = 32$  livelli di rosso e blu e 64 di verde, il bianco è rappresentato allora dalla terna (31,63,31), il rosso puro da quella (31,0,0) e così via. Si possono rappresentare in questo modo 65536 colori diversi, sufficienti nella maggior parte dei casi in quanto l'occhio è in grado di distinguerne alcune migliaia.

I grigi sono invece ottenuti per valori uguali del segnale emesso dai 3 fosfori e quindi, nel caso dell'esempio, si hanno 32 possibili livelli di grigio, fra i quali le terne (0,0,0), (1,2,1), (16,32,16) e (31,63,31), dove la prima e l'ultima terna corrispondono rispettivamente al nero e al bianco.

D'ora in avanti si parlerà quindi indifferentemente di numero di colori disponibili e di numero di bits per colore, tenendo presente che

**Numero di colori =  $2^n$  Numero di bits per colore**

Il principale problema correlato all'utilizzo della rappresentazione RGB è l'occupazione di memoria: una immagine di 640x480 pixels (la tipica risoluzione di una scheda VGA) occupa, a 16 bits per pixel, 600

Kbytes di memoria, mentre una 1024x768, che è uno dei formati emergenti ed è presente nelle schede Display Adapter 8514/A e XGA, occupa ben 1.5 Mbytes di memoria.

Il *frame buffer* (cioè lo spazio di memoria per le immagini) di una scheda video deve quindi essere molto grande, soprattutto se la scheda devono essere in grado di svolgere funzioni sofisticate di elaborazione delle immagini (esistono schede con 8 o anche 16 Mbytes di memoria) provocando una notevole dilatazione dei costi, particolarmente sensibile sui piccoli sistemi come i Personal Computers.

L'occupazione di memoria ha rilevanti conseguenze anche sui costi di archiviazione; tuttavia, in questo caso, si possono adottare tecniche di compressione che riducono in modo considerevole (con fattori 4 o anche superiori) lo spazio occupato da ciascuna immagine.

Sebbene esistano schede video a 16, ma anche a 24 bits per pixel (16 milioni di colori) come la scheda PC4000 Graphics Board realizzata dalla Real World Graphics LTD per Personal Computer o le schede delle Work-Station grafiche come la IBM-6090 o i RISC-6000, per le applicazioni non specializzate si adottano soluzioni di compromesso che comunque non portano gravi limitazioni.

### **I modelli a "look up table"**

Nella maggior parte dei casi il numero di colori contemporaneamente necessari per visualizzare una specifica immagine è nettamente inferiore al numero di colori necessari per visualizzare un insieme di immagini: questa osservazione viene sfruttata per i cosiddetti formati a "look up table" (o a tavolozza - LUT).

In questo caso per ogni immagine si seleziona un certo numero di colori (ad esempio 256) che ben la rappresentano, selezionandoli fra un insieme molto più ampio di possibili colori (ad esempio 65536): i colori selezionati costituiscono la tavolozza, o palette, dell'immagine e ogni pixel è descritto solamente dall'indice (da 0 a 255 - 1 byte - nell'esempio) dell'entry della palette che ne contiene il colore.

In questo modo una immagine 1024x768 occupa meno di 800 Kbytes (più 768 bytes per la palette) e quindi i files risultano di dimensioni più ridotte. Questa tecnica è spesso usata per gestire i frame buffers delle schede video, come l'adapter 8514/A, dove la palette è memorizzata in appositi registri detti "color registers". Modificando questi registri, tutti i pixels dello schermo relativi cambiano immediatamente colore, permettendo effetti di animazione ottenuti facendo ciclare rapidamente i colori (color cycling), come ad esempio per simulare lo scorrere dell'acqua.

### **I modelli del tipo "memory dump"**

Spesso, per ottenere il massimo in termini di velocità di visualizzazione, si sceglie di salvare una immagine come pura copia della memoria video della scheda (questo è ad esempio il caso del formato "bsave" del basic o delle funzioni getimage() e putimage() del Borland Turbo C). A parte problemi di prestazioni, questo è il modo peggiore che si possa scegliere, in quanto è strettamente legato alle specifiche hardware della scheda che si utilizza e non consente assolutamente la portabilità. Un tipico esempio di questo problema si riscontra con il programma StoryBoard Live, dove una presentazione realizzata per una risoluzione grafica (ad esempio 640x200, 4 colori) deve essere completamente riscritta per trasferirla ad una diversa risoluzione (ad esempio 640x480, 16 colori).

Il caso dello StoryBoard Live è però anche uno dei migliori esempi di come questa scelta di "non portabilità" sia stata necessaria per ottenere la possibilità di realizzare effetti di animazione, dissolvenze e esplosione di immagini sullo schermo a velocità accettabile anche su Personal Computer di limitate prestazioni.

Al contrario, una immagine realizzata in un formato indipendente dall'hardware come il PCX utilizzato da Paintbrush (e in pratica uno dei maggiori standard) sarà visualizzabile a qualunque risoluzione, anche se ovviamente con rese diverse.

## La compressione

La scelta del modello più opportuno è, per lo sviluppatore di un formato di files per immagini, solo il primo passo.

Il problema successivo è generalmente quello di ridurre l'occupazione di memoria su disco delle immagini. Esistono a questo scopo numerosi e diversi algoritmi di compressione che sono un ulteriore importante motivo di differenziazione fra i formati dei files (per di più, molti formati di files ammettono l'utilizzo di più di un algoritmo di compressione).

La scelta di uno fra i tanti è determinata da specifiche esigenze, dato che ciascuno ha, rispetto agli altri, specifici pregi: ad esempio, un maggiore fattore di compressione, una maggiore velocità di decompressione o un più facile accesso a uno specifico pixel dell'immagine.

Nota tecnica 2 - Algoritmi di compressione:

Esistono numerosi algoritmi di compressione per diminuire l'occupazione di memoria delle immagini e ciascun algoritmo si può presentare in diverse varianti per diversi formati di files grafici.

Le tecniche più moderne (ad esempio l'algoritmo J-PEG) consentono di ottenere considerevoli fattori di compressione (anche 20) a prezzo di una perdita di informazione rilevabile, per le immagini di tipo fotografico, solo da un osservatore esperto.

Tuttavia queste tecniche non sono attualmente implementate in formati di files standard e pertanto ci limiteremo a considerare solo gli algoritmi senza perdita di informazione più diffusi, che consentono di ottenere abbastanza facilmente fattori di compressione 2 o 3.

Alcune tecniche si adattano meglio ad immagini a colori, altre sono preferibili (o esclusive) per files in bianco e nero:

### PC Painbrush RLL

Questo è l'algoritmo utilizzato per la compressione dei files PCX. È un algoritmo orientato al byte, cioè comprime sequenze di bytes uguali. Ciascuna riga di pixels dell'immagine è compattata separatamente. Ciascun byte del file rappresenta un dato se il suo valore senza segno è minore o uguale a 192 (0xC0 esadecimale). In caso contrario è l'instestazione di un codice di ripetizione: i 6 bits meno significativi indicano quante volte deve essere ripetuto il dato che segue. Lo pseudo-codice per la decompressione è perciò riassumibile come segue:

```
Per ogni byte della linea leggi X
  Se X > 192, allora
    CONTATORE = (6 bits più bassi di X)
    DATO = (byte successivo a X)
  altrimenti
    CONTATORE = 1
    DATO = X
  ripeti DATO CONTATORE volte
```

### Packbits

Anche questo è un formato orientato al byte, nato in ambiente Macintosh e adottato per i files TIFF a colori e in diversi altri casi. Ciascuna riga è compattata separatamente. I dati sono memorizzati sotto forma di pacchetti preceduti da un byte di header; se il valore N (con segno) dell'header è compreso fra 0 e 127, allora seguono N+1 bytes di dati. Se N è compreso fra -127 e -1, il byte che segue deve essere ripetuto -N+1 volte. Se N = 128 non segue alcun dato. Lo pseudo codice può quindi essere:

```
Per ogni byte della linea leggi N
  Se 0 <= N <= 127 allora
    leggi N+1 bytes successivi
  altrimenti se -127 <= N <= -1
    CONTATORE = -N+1
    DATO = byte successivo
  ripeti DATO CONTATORE volte
```

### Compressione LZW

È un algoritmo piuttosto complesso utilizzato nei files GIF e dai programmi di compressione ARC e PKZIP. Funziona utilizzando un numero di bits superiore a 8 per rappresentare stringhe di bytes; in pratica i bits eccedenti vengono utilizzati per "estendere l'alfabeto" rappresentando stringhe di bytes con un unico codice di più di 8 bit. Con 9 bits ad esempio si hanno a disposizione 256 codici su 512 (gli altri 256 sono utilizzati per descrivere i bytes semplici) per codificare stringhe opportunamente costruite.

### CCITT gruppo 3

È un algoritmo specifico per immagini in bianco e nero applicato nei files TIFF e per la trasmissione di immagini FAX. È basato su una codifica di Huffman modificata; utilizza cioè codici di lunghezza variabile per rappresentare sequenze (run) di bits uguali. Mediante una analisi statistica si sono determinate le frequenze con le quali si incontrano nelle immagini run di diversa lunghezza

e si sono assegnati a tali run i codici più corti. I codici sono costruiti con una struttura ad albero che permette di implementare algoritmi di decompressione molto efficienti.

**CCITT gruppo 4 (MMR)** È una evoluzione del gruppo 3, che migliora il fattore di compressione ottenibile codificando i run di una linea (dove conveniente) per differenza rispetto alla linea precedente. Per maggiori dettagli riferirsi alle fonti citate in Bibliografia.

## Il descrittore

Se per un formato di tipo "memory dump" non è necessario conoscere altro che la scheda e la risoluzione a cui si riferisce, ben diversa è la situazione per un file hardware independent.

In questo caso i dati relativi ai pixels (che potremmo generalmente chiamare la "pixmap") devono essere accompagnati da una completa descrizione delle dimensioni della pixmap stessa (numero di righe e colonne), del numero di colori, dell'eventuale algoritmo di compressione, della palette e di quant'altro necessario: il cosiddetto descrittore dell'immagine.

È evidente che anche questa descrizione può essere fatta in molti modi, posizionata su un file separato o precedere o seguire i dati della pixmap nello stesso file: molti formati, per i quali i dati sono rappresentati nello stesso modo, differiscono in effetti proprio per il descrittore.

## Il problema dei colori

Sebbene sia corretto affermare che per un formato indipendente dall'hardware una immagine è visualizzabile a qualunque risoluzione, questo tuttavia non garantisce la qualità del risultato. In particolare ciò è vero quando, data una immagine acquisita (ad esempio con una telecamera) a un numero elevato di bits per pixel (ad esempio 24), la si vuole visualizzare con un numero ridotto di colori (256 o addirittura 16). In questo caso è necessario eseguire sull'immagine un pre-processing che ne riduca il numero di colori applicando particolari classi di algoritmi (noti come dithering e error diffusion) che ne riducano il numero di colori nel modo migliore possibile simulando, con l'accostamento di pixels diversi, i colori intermedi non ottenibili.

In generale gli algoritmi di error diffusion producono risultati migliori ma sono sensibilmente più lenti di quelli di dithering.

Un altro aspetto da tener presente quando si applicano questi algoritmi riguarda la palette da utilizzare: si può infatti cercare per ogni immagine una palette ottimizzata, ma questo impedisce di sovrapporre più immagini sulla stessa schermata, e quindi se si vuole sfruttare questa possibilità bisogna adottare una palette "standard", cioè comune a tutte le immagini.

In particolare questo è vero per il caso dei 16 colori, che corrispondono generalmente all'utilizzo di immagini su schede grafiche EGA o VGA per le quali esiste una palette di default, che spesso non è possibile modificare.

Per le immagini a 16 colori non ci si debbono comunque aspettare risultati di sufficiente qualità nel caso di fotografie digitalizzate, poiché in tal caso le informazioni a disposizione sono obiettivamente insufficienti (in particolare ad esempio il color carne viene rappresentato da pixels bianchi con qualche macchia rossa e si hanno colori molto intensi e contrastati).

Nota tecnica 3 - Tecniche di riduzione dei colori.:

Le tecniche utilizzate per ridurre il numero di colori di una immagine sono generalmente 4:

**Troncamento** Il metodo più rapido per ridurre il numero di colori consiste nel troncamento dei bits meno significativi (ogni bit eliminato dimezza il numero di colori). Questo metodo è molto semplice ed efficiente, tuttavia i risultati sono spesso scadenti. È consigliabile per portare immagini da 24 a 16 bits, poiché l'occhio distingue con facilità solo alcune migliaia di colori e quindi, in questo caso, non risente in modo significativo del troncamento.

<b>Compattazione</b>	Consiste nel selezionare un insieme di colori (generalmente i più frequenti nell'immagine, ottenuti mediante istogrammazione) e nel fare convergere su questi tutti gli altri, scegliendo per ciascuno quello con distanza cartesiana (nello spazio RGB dei colori) minore. Per una applicazione di questa tecnica, vedere l'articolo di D.Pomerantz citato in bibliografia. La tecnica è spesso sufficiente per visualizzare immagini RGB a 16 bits su schede grafiche con palette da 256 colori su 16 milioni. Il metodo non è però utilizzabile se è necessario servirsi di una palette standard.
<b>Dithering</b>	Questa tecnica consiste nello sfruttare il fatto che l'occhio tende a fondere il colore di pixels vicini generando l'impressione di un unico colore intermedio. I colori non ottenibili direttamente sono realizzati accostando in patterns predefiniti pixels dei colori presenti nella palette (fissa o costruita per istogrammazione). Esistono algoritmi per implementare in modo molto efficiente questa tecnica. Il principale problema consiste nel fatto che, per superfici estese dello stesso colore, risulta molto evidente all'occhio il reticolo regolare che costituisce il pattern, creando nell'osservatore una sensazione di "artificialità" dell'immagine.
<b>Error diffusion</b>	Sfrutta lo stesso effetto ottico del dithering, ma invece di realizzare patterns regolari i pixels sono accostati utilizzando una tecnica di propagazione degli errori; il valore di ciascun pixel è scelto in modo da compensare per quanto possibile gli errori effettuati per quello precedente e per il corrispondente della riga superiore. È la tecnica migliore, anche se difficile da implementare in modo efficiente. Può essere utilizzata per portare immagini a 256 o anche a 16 colori.

## La raccolta di routines

Da quanto detto sino ad ora appare evidente come sia impraticabile l'idea di realizzare un unico tool che, dati i formati dell'immagine di partenza e di quella di arrivo, effettui automaticamente la conversione, se non altro perché tale strumento diventerebbe rapidamente obsoleto.

La soluzione migliore, a patto di disporre di una discreta familiarità con la riga comandi del DOS, consiste nel disporre di un insieme il più ampio possibile di tools che eseguano specifiche conversioni e nel comporli l'uno con l'altro fino ad ottenere il risultato desiderato.

Ogni nuova esigenza può così essere soddisfatta aggiungendo un solo programma mentre una complessa catena di conversioni può essere automatizzata da un file comandi (file di tipo .BAT), come sarà mostrato in alcuni esempi.

Questo lavoro può essere notevolmente semplificato dalla scelta di un formato intermedio particolarmente semplice utilizzato come tipo sorgente o destinazione da tutte le routines di conversione: in questo modo l'introduzione di un nuovo formato si riduce alla realizzazione di un solo programma (o 2, se si preferisce tenere separate le conversioni nei 2 sensi) da e verso lo standard intermedio.

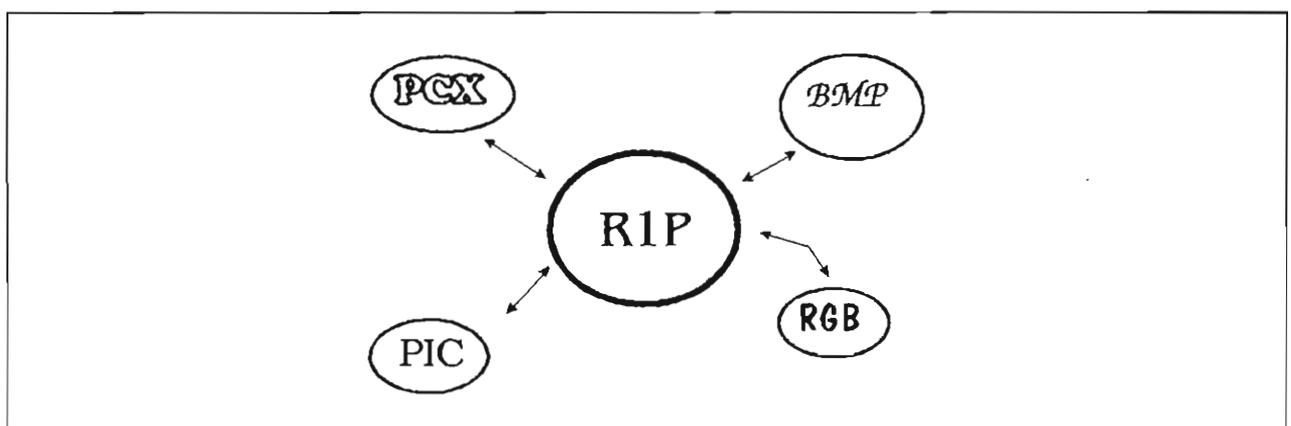


Figura 4. Lo schema utilizzato per le conversioni.

## Il formato intermedio R1P

Il formato intermedio scelto è lo stesso utilizzato dalle citate PCPAINTBRUSH Utilities e da diversi altri pacchetti di uso interno: il cosiddetto formato R1P.

Questo formato è assai semplice, in quanto non è altro che una immagine raster non compressa seguita da un descrittore.

La scelta di mettere il descrittore in fondo all'immagine è motivata dal fatto che in questo modo i programmi di lettura dei dati grafici funzionano come se il descrittore stesso non esistesse e possono trattare il file come una pixmap pura. In realtà, il descrittore stesso è opzionale e può essere omesso o scritto in un file a parte (detto di tipo DSC).

È importante osservare che il fatto di avere le informazioni di descrizione del file in coda non comporta alcun problema nel caso di utilizzo del linguaggio C o comunque di linguaggi che dispongano di istruzioni di accesso casuale ai files; in caso di accesso sequenziale, tuttavia, per raggiungere queste informazioni sarebbe necessario scorrere l'intero file dati per raggiungere il descrittore.

Nella quasi totalità dei casi considerati i files R1P sono relativi a immagini a 256 livelli di grigio o a 256 colori con look up table, e pertanto la pixmap si riduce a una bytemap, cioè ogni pixel è rappresentato esattamente da un byte: questo rende ancora più semplice l'utilizzazione dei dati poiché le operazioni di input-output risultano elementari.

Quanto segue è una descrizione dettagliata del formato, utile a chi voglia realizzare proprie routines di conversione.

### I dati raster

La pixmap è costituita da un singolo piano di **yy** linee dell'immagine, organizzate dall'alto al basso, ciascuna lunga **bpl** bytes. Ciascuna linea contiene **xx** pixels organizzati da sinistra a destra (ordine lessicografico) che sono descritte da un valore costituito da **bpp** bits. Il pixel più a sinistra della linea si trova sul byte di indirizzo più basso e a partire dal bit più significativo verso quelli meno significativi. Il valore degli eventuali bits che vanno dal pixel più a destra sino alla fine dell'ultimo byte della linea, come definito da **bpl**, non è determinato.

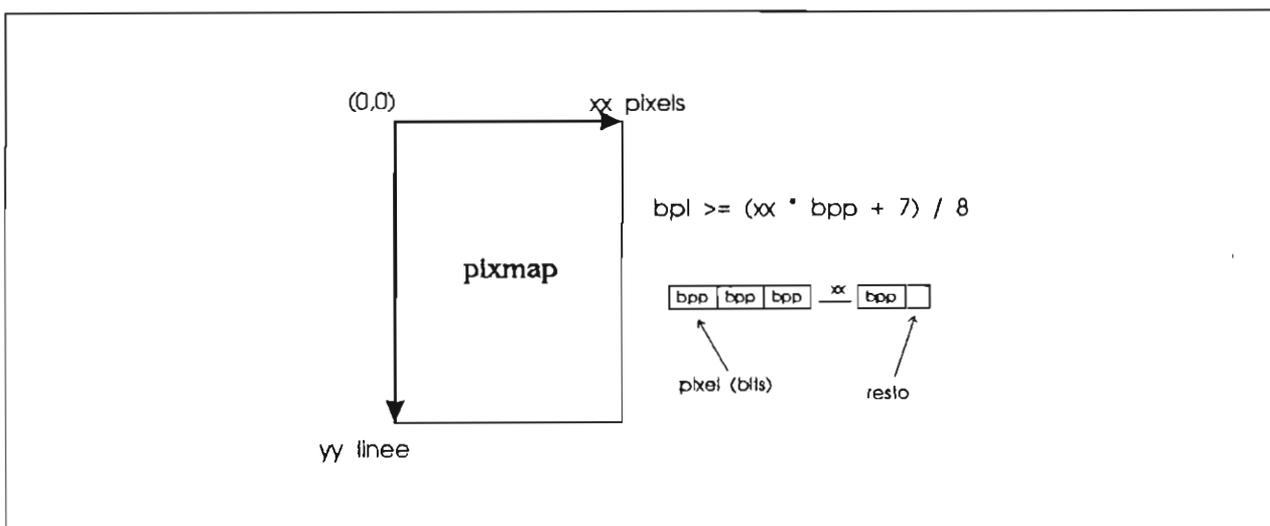


Figura 5. I parametri di una immagine raster

Sebbene questa descrizione possa a prima vista sembrare complicata, corrisponde all'organizzazione più intuitiva.

## Il descrittore

Il descrittore, posto alla fine del file, è sempre costituito da almeno 8 bytes (che chiameremo trailer), raggruppati in 3 interi a 2 bytes in rappresentazione *little endian* e 2 bytes separati.

Nota tecnica 4 - La rappresentazione dei numeri:

Uno dei principali problemi connessi con la portabilità dei programmi e dei files di dati fra diverse piattaforme hardware consiste nel modo utilizzato dalle diverse CPU di ordinare in memoria i bytes delle **words** (coppie di bytes, usate ad esempio nei Personal Computer per i numeri interi) e delle **double words** (insiemi di 4 bytes, usati nei Personal Computer per gli interi lunghi). Vediamo la rappresentazione utilizzata per le macchine più diffuse:

### word

**little endian** I processori INTEL utilizzati nei PS/2 IBM, ma anche i VAX e i PDP-11 della Digital, utilizzano un formato detto *little endian* nel quale viene memorizzato per primo (ossia in locazioni di memoria inferiori) il byte meno significativo e per secondo quello più significativo:

lo	HI
----	----

e quindi, ad esempio, il numero intero senza segno 1000 (il cui byte meno significativo vale 232 e quello più significativo vale 3, infatti  $1000 = 232 + 3 \cdot 256$ ) è rappresentato in memoria come segue:

232	3
-----	---

**BIG endian** I processori MOTOROLA della serie 68k, utilizzati ad esempio sulla Work-Station grafica IBM 6090 e sugli Appie Macintosh, e i sistemi IBM Risc 6000 o gli host IBM della serie S/370, utilizzano un formato detto *BIG endian* nel quale viene memorizzato per primo (ossia in locazioni di memoria inferiori) il byte più significativo e per secondo quello meno significativo:

HI	lo
----	----

e quindi, ad esempio, il numero intero senza segno 1000 dell'esempio precedente viene rappresentato in memoria come segue:

3	232
---	-----

### double word

**little endian** I processori INTEL e i VAX utilizzano anche per le **double words** il formato *little endian* nel quale viene memorizzata per prima la **word** meno significativa con il byte meno significativo per primo:

lo word		HI word	
lo	HI	lo	HI

**BIG endian** I processori MOTOROLA della serie 68k, i sistemi IBM Risc 6000 e gli host IBM della serie S/370, utilizzano il formato *BIG endian* nel quale viene memorizzata per prima la **word** più significativa con il byte più significativo per primo:

HI word		lo word	
HI	lo	HI	lo

**PDP-11** Il PDP-11 utilizza una rappresentazione mista, in cui le **word** sono disposte in ordine *BIG endian*, ma i bytes di ciascuna **word** sono in ordine *little endian*:

HI word		lo word	
lo	HI	lo	HI

**little BIG** L'ultima possibile configurazione, prevede di avere le word in ordine little endian, con i loro bytes costitutivi in ordine BIG endian:

lo word		HI word	
HI	lo	HI	lo

I 5 campi descritti definiscono (da sinistra a destra) **xx**, **yy**, **bpl**, **bpp** e **stato**.

<b>xx</b>	<b>yy</b>	<b>bpl</b>	<b>bpp</b>	<b>stato</b>
-----------	-----------	------------	------------	--------------

Se il byte di stato indica che è presente una palette, questa precede il trailer nel file.

Se il byte di stato indica che è presente un commento, questo precede la palette (se presente) e il trailer nel file.

Gli 8 bits del byte di **stato** sono codificati come segue:

<b>PP</b>	<b>C</b>	<b>N</b>	<b>xxxx</b>
-----------	----------	----------	-------------

Dove:

**PP** (i due bits più significativi) costituiscono l'indicatore della palette e assumono il seguente significato:

**00** Immagine di tipo "look up table" con palette esterna in un file di tipo RGB (ossia 1 byte di identificazione contenente il carattere esadecimale 0x0C seguito da tante terne di bytes, in formato R-G-B con intensità da 0 a 255, quanti sono i colori presenti nella lookup table, cioè  $2^{\text{bpp}}$ ).

Il default per immagini monocromatiche è 0=nero, 1=bianco; per immagini a 16 colori è la palette standard EGA, per immagini a 256 colori la palette di default del programma PC Paintbrush.

**01** Immagine a toni di grigio con livelli da nero = 0 a bianco =  $2^{\text{bpp}} - 1$ .

**10** Immagine di tipo "look up table" con palette interna in formato RGB. La palette precede immediatamente il trailer nel file ed è composta dal byte di identificazione 0x0C e da  $2^{\text{bpp}}$  entries di 3 bytes con le intensità R G e B di ciascun colore (fra 0 e 255). In pratica si incontrano solo palettes a 2, 16 e 256 entries corrispondenti a 1, 4 e 8 **bpp**.

**11** Immagine di tipo RGB. Dati i **bpp** bits relativi a ogni pixel, il numero di bits per il Red e per il Blu è la parte intera di  $\text{bpp}/3$ , mentre i bits restanti sono assegnati al Green.

**C** è l'indicatore di commento e deve essere posto a 1 se esiste un commento. Questo deve precedere la palette (**PP** = 10) o il trailer se questa non esiste (**PP** = 00,01,11). Il commento deve essere una stringa di caratteri ASCII delimitati da bytes nulli (cioè posti a 0) e terminata con i caratteri di CR (carriage return) o CRLF (carriage return + line feed).

**N** è l'indicatore di normalizzazione. È posto a 1 se i dati devono essere rinormalizzati, cioè se il valore minimo e massimo presenti devono essere riportati a 0 e al valore massimo possibile, riscalando tutti i valori intermedi.

## Il descrittore esterno

Diverse routines e programmi sviluppati presso i Centri Ricerca di Milano e Roma dividono i dati grafici, il descrittore e la palette su files separati.

Sebbene i dati grafici siano identici, la palette e il descrittore sono diversi da quelli per il formato RIP con descrittore interno.

La palette è salvata in un file di 1024 bytes ( di estensione .PAL) costituito da 256 entries da 4 bytes R-B-G- $\Pi$  (ATTENZIONE: non R-G-B- $\Pi$ ), dove l'ultimo byte non ha alcun significato, ma è presente in modo che ciascuna entry sia esattamente delle dimensioni di una double word; questo formato è utile poiché è lo stesso utilizzato per i registri di colore del display adapter 8514/A, per il quale le routines sono appositamente realizzate (il programma **CPY8514** permette la conversione fra i formati di palette PAL e RGB).

Per quanto riguarda il descrittore, il file DSC è utilizzato non solo per descrivere immagini raster del tipo dei files RIP a 8 bpp (che vengono qui chiamate immagini FRAME) ma anche per immagini di tipo RGB dove i dati relativi a ciascuna delle 3 bande vengono memorizzati in tre diversi files. Ciascun file contiene la pixmap di una banda a 8 bpp (immagini qui dette di tipo BAND). Si memorizzano in questo modo immagini a 24 bits per pixel. Questo formato è utilizzato per memorizzare le immagini acquisite con telecamera o mediante lo scanner di diapositive Eikonix 1435 Slide Scanner prodotto dalla Kodak Eikonix Imaging Systems, nonché da numerosi programmi in ambiente mainframe IBM.

Il file descrittore è costituito da

- 3 records di 82 caratteri per immagini FRAME
- 5 records di 82 caratteri per immagini BAND (RGB su tre bande)

Esaminiamo uno a uno i diversi records, utilizzando la seguente notazione:

<b>nn..mm</b>	indica che i byte da nn a mm sono posti a 32 (il codice ASCII dello spazio) oppure al valore indicato a fianco (ad esempio 0..7 imgname indica che i bytes da 0 a 7 contengono il nome dell'immagine)
<b>'x'</b>	indica che il byte corrispondente contiene il carattere ASCII x
<b>0xhh</b>	indica che il byte corrispondente contiene il codice esadecimale 0xhh
<b>CCC</b>	indica che i byte corrispondenti contengono la stringa di caratteri ASCII 'CCC'
<b>!</b>	questo carattere separa logicamente i vari campi del record (non compare come effettivo carattere nel record)

Sulla base di queste convenzioni i records possono essere descritti come segue:

Per solo uso interno IBM

Record 1 -

```
! 0..79 ! '1' 0x0d 0x0a
  commento
```

Record 2 -

```
FRAME image (con lookup table) - su FG4000 è una BW image
! 0..27 !FRAME! 33..35!xxxxx!xxxxx!46..48!'1'!50..52!'0'!54..78! '1' 0x0d 0x0a
  -----      xsize ysize      -
3 BANDs Color Image
! 0..27 !BAND! 32..35 !xxxxx!xxxxx!46..48!'3'!50..52!'0'!54..78! '1' 0x0d 0x0a
  ----      xsize ysize      -
Dove '1' e il 3 sottolineati indicano il numero di records che
seguono.
```

Record 3 -

```
FRAME image (con lookup table) - su FG4000 è una BW image
! 0..7 imgname !RGB! 11..16 ! '8' '3' !20..29 !IDB$PAL$!PAL! 41..78 ! '1' 0x0d 0x0a
                                     !BW      !
                                     !imgname !

3 BANDs Color Image
! 0..7 imgname !RED! 11..16 ! '8' '3' !20..78 ! '1' 0x0d 0x0a
```

Records 4 e 5 (solo nel caso di Color image)

```
! 0..7 imgname !GRE! 11..16 ! '8' '3' !20..78 ! '1' 0x0d 0x0a
! 0..7 imgname !BLU! 11..16 ! '8' '3' !20..78 ! '1' 0x0d 0x0a
```

dove **imgname** è il nome del file (o dei files) contenente la pixmap e **xsize** e **ysize** i numeri di righe e colonne dell'immagine.

Il programma **MAKEDSC** permette di scrivere un descrittore passando a parametri le caratteristiche del file che deve descrivere. Sono inoltre allegati i sorgenti di routines in C per scrivere e leggere sia i descrittori interni ai files RIP che quelli esterni.



## Esempi di conversioni di formato

I paragrafi seguenti mostrano alcuni esempi di conversioni fra formati grafici, descritti passo passo. Per collegare questi esempi a un reale ambiente di sviluppo si è scelto di utilizzare come modello la configurazione hardware e software presente al Centro Ricerca di Milano, di cui è opportuno descrivere preliminarmente l'architettura. Per informazioni sull'origine e la reperibilità delle routines citate, riferirsi alla tabella in Appendice F, "Utili prodotti software e TOOLS" a pagina 53.

## Strumenti di acquisizione

Presso questo Centro sono attualmente disponibili diversi dispositivi di acquisizione di immagini a colori e in bianco e nero collegati a Personal Computers:

- La scheda **VCA (Video Capture Adapter) IBM** permette di acquisire immagini a colori da telecamera a 16 bits per pixel. Il programma **AVC** può salvare immagini così acquisite in un formato detto **RGBImage** (files con estensione **\_ID**): tali files sono costituiti da un header di 880 bytes (la cui descrizione è contenuta nel file **AVCFILES.CNC** presente sui dischetti di **AVC**), seguito dalla bytemap dell'immagine; ogni pixel è rappresentato da 2 bytes in formato RGB 5 bits red, 6 green, 5 blue.

Il programma può funzionare sia in ambiente OS/2 che DOS.

- La coppia di schede **PC4000 Graphics Board** e **FG4000 Frame Grab** realizzate dalla Real World Graphics LTD consente acquisizioni a 24 bpp su Personal Computer con bus di tipo AT. Attualmente le acquisizioni vengono effettuate utilizzando il programma **TVCAMER** realizzato presso il Centro Ricerca di Roma. Tale programma memorizza su disco le immagini a colori in tre distinte bytemap per le bande rossa, verde e blu (files con estensione **RED**, **GRE** e **BLU**), più un descrittore di tipo **DSC**. In alternativa è possibile ottenere immagini a 256 livelli di grigio.
- Lo scanner per diapositive **Eikonix 1435 Slide Scanner** prodotto dalla Kodak Eikonix Imaging Systems permette di acquisire diapositive a colori a 24bpp o in bianco e nero a 256 livelli di grigio. Si utilizza il programma **ICONA**, realizzato presso il Centro Ricerca di Milano, che memorizza su disco le immagini nello stesso formato di **TVCAMER** per acquisizioni a colori e come una bytemap più descrittore (di tipo **FRAME**) per immagini in bianco e nero.
- Lo scanner **IBM 3119 Page Scanner** permette di acquisire immagini sino al formato A4 in bianco e nero o a 256 livelli di grigio. Il semplice programma fornito a corredo dello scanner permette di acquisire immagini e salvarle in formato **TIFF** (uno dei formati più utilizzati nel campo dell'editoria elettronica).
- L'**IBM 3118 Scanner** permette di acquisire immagini formato A4 in bianco e nero ed è collegato ad un host **VM/CMS**.

Vi sono anche diversi scanners di grosse dimensioni (fino al formato A0) utilizzati per applicazioni relative al campo dei disegni tecnici e che non vengono qui considerati, poiché l'interesse di questo lavoro è rivolto ad applicazioni di tipo raster e non vector.

## Utilizzo delle immagini acquisite

Le immagini acquisite con i dispositivi elencati possono venire utilizzate in diversi modi, come ad esempio:

- All'interno di programmi sviluppati presso i Centri Ricerca (generalmente in questo caso è utilizzato il formato **RIP**).
- Per la preparazione di presentazioni in **Storyboard Live** nei diversi formati a 16 e 256 colori.
- Per applicazioni di elaborazione di immagini su **PS/2**, **Risc6000** e **VM** (formati **RIP** e **RGB** in 3 bande separate).
- Stampa a colori o in bianco e nero dei risultati di elaborazioni di immagini (formati **38PSEG**, **RGB** o comunque visualizzabili su schermo per le **Hard-Copiers** a colori).

- Operazioni di editing sull'immagine con programmi del tipo di PaintBrush + ® (formato PCX).

Presso il Centro Ricerca di Milano sono installati numerosi dispositivi per la visualizzazione e la stampa delle immagini, sia in ambiente PS/2 che RISC6000 e VM:

- **Schede grafiche per PS/2**

- **PS/2 Display Adapter** - La scheda grafica standard installata su tutti i PS/2.
- **PS/2 Display Adapter 8514/A** - Scheda grafica ad alte prestazioni per PS/2, con una risoluzione massima di 1024x768 pixels e 256 colori su 16 milioni.
- **PS/2 XGA Display Adapter** - Scheda per PS/2 di ultima generazione, rappresenta una evoluzione della 8514/A rispetto alla quale presenta notevoli miglioramenti (è ad esempio presente un modo di indirizzamento diretto nel quale ogni pixel -a una risoluzione di 640x480- può avere un colore indipendente scelto fra 65536 colori -16 bits- in coordinate RGB).

- **Sistemi di visualizzazione in ambiente RISC6000**

- **Adattatore Grafico a Colori** - Scheda entry level, con una risoluzione di 1280x1024 e 256 colori contemporaneamente su 16 milioni.
- **Adattatore Grafico 3D** - Scheda ad alte prestazioni con frame buffer a 24 bit, capace pertanto di visualizzare immagini in formato RGB a 24 bits per pixel, con una risoluzione massima di 1280x1024 pixels.
- **X-Station 120** - È un terminale intelligente specializzato nel trattare applicazioni grafiche basate su X-Windows. Le capacità grafiche dipendono dalla memoria installata e dal monitor utilizzato, permettendo di arrivare sino ad una risoluzione di 1280x1024 con 256 colori.

- **Sistemi di visualizzazione in ambiente VM**

- **Workstation Grafiche IBM 6090 e IBM 5080** - Sono stazioni di lavoro grafiche molto potenti, specializzate per l'uso in ambiente graPHIGS. In particolare la IBM 6090 dispone di un frame buffer a 24 bit con una risoluzione di 1280x1024.
- **Terminali grafici 3192** - Sono i comuni terminali grafici GDDM.

- **Stampanti**

- **IBM 4019** - Stampante laser in bianco e nero con modulo opzionale PostScript. Permette anche la stampa in emulazione HP-LaserJet ed ha una risoluzione massima di 400 dpi (dots per inch). È collegabile via porta parallela a PS/2 e RISC6000.
- **IBM 3812** - Stampante laser in bianco e nero in ambiente VM con una risoluzione massima di 240 dpi.
- **IBM 6097 Screen Printer** - Hard copier a colori collegabile direttamente al monitor IBM 6091 e utilizzabile su RISC6000 e IBM 6090 o 5080. Permette di ottenere stampe di alta qualità su carta speciale o trasparenze, implementando in hardware tecniche di dithering.
- **IBM 5087 Screen Printer** - Hard Copier a colori, modello meno sofisticato della 6097.
- **IBM Proprinter XL III** - Stampante ad aghi convenzionale per PS/2 e RISC6000.

## Acquisizione di una immagine mediante FG4000 e display su 8514/A

Il programma di acquisizione TVCAMER per la schede FG4000-PC4000 produce 4 files: le tre bande con estensioni RED, GRE e BLU rispettivamente più il descrittore DSC. Da questi files bisogna ottenerne uno unico in formato RIP con palette ottimizzata sull'immagine e descrittore interno al file stesso.

La prima operazione da compiere è quella di ridurre il numero di colori per portare l'immagine da 24 bpp a 8 bpp. Questa operazione può essere compiuta con la routine **RGBCONV**, che assume che le estensioni dei files delle tre bande siano R, G e B. Inoltre è necessario conoscere il numero di pixels per ogni riga dell'immagine (desumibili dalla lettura del descrittore).

Per solo uso interno IBM

La routine **CRGBCONV** (Call RGBCONV) si occupa di invocare quest'ultima leggendo i dati necessari dal descrittore.

Il formato in uscita è di tipo AFI TARGA1

I comandi da digitare al prompt del DOS sono pertanto (se fname è il nome dell'immagine):

```
ren fname.RED fname.R
ren fname.GRE fname.G
ren fname.BLU fname.B
```

```
CRGBCONV fname
```

Il passo successivo consiste nell'utilizzare la routine **VGA2PMBM** per convertire il file ottenuto in formato BMP (il formato nativo di OS/2, noto anche come PMBM - Presentation Manager Bit Map) con i comandi

```
VGA2PMBM fname.AFI fname.BMP
del fname.AFI
```

Infine si procede alla conversione in formato R1P:

```
PMBM2R1P fname.BMP fname.R1P
del fname.BMP
```

Il file ottenuto ha tutte le caratteristiche richieste e può essere visualizzato su 8514/A con il programma **R1P28514**.

La procedura descritta per arrivare al file in formato R1P è raccolta nel batch file **FG2R1P.BAT**.

## Utilizzo di un file R1P in uno stampato SCRIPT o BOOKMASTER

Il Personal Computer è l'ambiente ideale per realizzare in modo semplice e rapido illustrazioni e charts da inserire in pubblicazioni; tuttavia, soprattutto per gli usi interni IBM, gli stampati devono essere realizzati in ambiente **SCRIPT** o **BOOKMASTER**, che sono i prodotti standard di formattazione testi sui sistemi VM.

Sorge pertanto la necessità di utilizzare immagini realizzate su Personal Computer all'interno di questi pacchetti.

Per ottenere questo risultato, a partire da una immagine in formato **R1P**, il formato intermedio, occorre prima di tutto ridurre l'immagine ad una bitmap., cioè a due soli colori.

Se l'immagine di partenza è un file R1P a 4, 16 o addirittura 256 colori, la riduzione a 2 soli livelli di grigio deve essere eseguita per lo meno utilizzando un algoritmo di dithering; in caso contrario i risultati ottenuti saranno insufficienti.

Per effettuare questa prima conversione occorre utilizzare la routine **R1PCLIP**, ad esempio:

```
R1PCLIP source.R1P destfile.R1P G
```

porta da 16 colori a bianco e nero utilizzando una matrice di dithering 2x2, raddoppiando, quindi, le dimensioni effettive dell'immagine (l'opzione R a sua volta permette di dimezzarne le dimensioni riportandole a

quelle originali, ma si possono ottenere disultati indesiderati, poiché la matrice di dithering è regolare e mal si presta a riduzioni per eliminazione di pixels).

Una volta ottenuta una immagine *bilevel*, questa deve essere portata in formato **PCX** utilizzando **RIP2PB**:

```
RIP2PB source.RIP destfile.PCX
```

Il file così ottenuto deve essere convertito in formato **PSEG38PP**; questo richiede due passi: uno eseguito sul Personal Computer, l'altro in ambiente VM. La prima elaborazione è svolta da **PB1T38PP**:

```
PB1T38PP source.PCX destfile.PP
```

L'immagine generata in questo modo verrà stampata utilizzando una risoluzione di 120 punti per pollice; questo significa che le dimensioni reali dell'immagine, stampata su IBM 3812, saranno date (esprese in pollici) dal numero di pixels diviso per 120.

È però possibile (e consigliabile) specificare l'opzione **H**, che genera immagini a 240 punti per pollice, molto migliori delle precedenti.

Il file deve quindi essere trasferito sull'host VM in forma binaria, ad esempio utilizzando il comando:

```
mytecopy source.PP H:destfile PPBIN /B /V=256 /S
```

e convertito nella sua forma definitiva per mezzo del comando:

```
PPBIN2PSE source PPBIN a destfile PSEG38PP a
```

Il file generato può essere incluso (*imbedded*) in un programma **SCRIPT** per mezzo delle istruzioni:

```
:fig id=fnnn frame=box
.ce on
.si filename inline
.ce off
:efig
```

oppure in un elaborato **BOOKMASTER** con :

```
:fig id=fnnn frame=box
:artwork name=filename align=center.
:efig
```

In fase di compilazione è poi necessario fornire al comando **SCRIPT** le opzioni **SYSVAR(G INLINE)** e **SEGLIB(PSEG38PP)**.

## Utilizzo di una diapositiva in una presentazione StoryBoard Live

Per inserire una diapositiva in una presentazione realizzata con StoryBoard Live, occorre prima di tutto acquisirla in forma digitale; lo scanner per diapositive **Eikonix 1435 Slide Scanner**, pilotato dal programma **ICONA** permette di ottenere immagini di ottima qualità a 24 bits per pixel.

Il programma icona genera tre files in formato RGB a tre bande separate con un byte per banda più un descrittore; l'immagine ottenuta deve quindi essere ridotta a 8 bits per pixel (256 colori).

A questo scopo occorre utilizzare la routine **RGBCONV2** con il comando DOS:

```
RGBCONV2 file newfile
```

Il file generato è in formato R1P con descrittore esterno e può essere visualizzato su schermo utilizzando **R1P28514** o **R1P2VGA** e quindi catturato con il programma **PT Picture Taker** fornito assieme al prodotto StoryBoard Live.

Questa procedura funziona anche con le versioni di Story Board precedenti alla Live; in alternativa, in questo ultimo caso, è possibile portare l'immagine in formato PCX e quindi importarla direttamente nel modulo **PM - Picture Maker** di StoryBoard Live:

```
R1P2R1P -i newfile.R1P newfile.R1P  
R1P2PB newfile.R1P newfile.PCX
```

## Realizzazione di una "trasparenza" a colori per una presentazione.

In ambiente Personal Computer sono disponibili diversi programmi, molto semplici da utilizzare, per realizzare "trasparenze" per presentazioni, posters per conferenze e simili; ad esempio, molto diffuso è il programma **PaintBrush +** ®, distribuito come accessorio standard di Microsoft Windows 3.

I files in formato PCX realizzati con questi programmi possono essere stampati a colori con ottimi risultati utilizzando la **Hard-Copier** a colori **IBM 6097** sia in ambiente VM con una stazione grafica **IBM 6090** che utilizzando un **RISC 6000**.

A questo scopo è sufficiente convertire l'immagine in formato a 3 bande con un byte per banda per pixel utilizzando i seguenti comandi:

```
PB2R1P fname.PCX fname.R1P  
R1P2RGB fname
```

I files così generati possono poi venire trasferiti su VM utilizzando, ad esempio, **MYTE**:

```
mytecopy fname.RED H:fname RED fm /B /lrecl=lsiz  
mytecopy fname.GRE H:fname GRE fm /B /lrecl=lsiz  
mytecopy fname.BLU H:fname BLU fm /B /lrecl=lsiz
```

dove *lsiz* è la lunghezza in bytes di una linea dell'immagine, reperibile nel descrittore di tipo DSC creato da **R1P2RGB**.

L'immagine può essere visualizzata utilizzando la routine **SHI6090** e stampata dopo averne eventualmente corretto i livelli medi di colore.

In alternativa, trasferendo l'immagine in ambiente RISC (o utilizzando dischetti e il comando *DOSREAD dosfname aixfname* o mediante **FTP**), è possibile visualizzarla utilizzando il programma **BJPIC** e stamparla con la Hard-Copier.

## Le routines utilizzate: manuale di riferimento

Quella che segue è una descrizione delle routines citate nel presente lavoro. I programmi sviluppati presso il Centro Ricerca IBM di Milano, e non documentati altrove, sono descritti in maniera completa, mentre per gli altri è data solo una descrizione sommaria, anche se sufficiente per il normale utilizzo. Si rimanda alla documentazione specifica per ulteriori informazioni.

Per quanto riguarda le routines non sviluppate presso i Centri Ricerca di Milano e Roma, il campo *Fonte* riporta il prodotto o il tool cui appartengono. Per maggiori dettagli sulla reperibilità riferirsi alla Appendice F, "Utili prodotti software e TOOLS" a pagina 53.

### 5652BAND

#### *Sintassi*

5652BAND <opzioni> filename <.ext>

#### *Ambiente*

DOS

#### *Descrizione*

Il programma esegue la conversione di files di tipo RGB 565, cioè a 16 bits per pixel (65536 colori), di cui 5 per il rosso, 6 per il verde e 5 per il blu, in tre files separati a un byte per banda (3x8 = 24 bits per pixel).

Vengono generati 3 files di estensione RED, GRE e BLU e un descrittore di tipo DSC.

Se viene passato il solo filename, il programma assume si tratti di un file di tipo RGBImage (con estensione \_ID), acquisito con il programma AVC e la scheda VCA. Tali files sono costituiti da un header di 880 bytes (dal quale la routine ricava le informazioni sulle dimensioni) e da una pixmap a 16 bpp, 5 bits red, 6 green e 5 blu.

Altrimenti occorre passare al programma le dimensioni dell'immagine in pixels e l'offset dal quale partono i dati; questo consente di saltare un eventuale header.

La conversione da 5 (o 6) bits per banda a 8 viene effettuata mediante uno shift a sinistra di 3 (o 2) posizioni.

#### *Opzioni*

-o nn	offset dal quale partono i dati nel file (in bytes)
-w nn	ampiezza in pixels dell'immagine
-h nn	altezza in pixels dell'immagine

#### *Fonte*

Centro Ricerca IBM di Milano

## AV2R1P

### *Sintassi*

AV2R1P filename

### *Ambiente*

DOS

### *Descrizione*

Questo batch file accetta il pathname completo di un file `_ID` (passato senza estensione) e dopo alcuni passaggi intermedi lo converte in un file formato RIP contenente una palette ottimizzata. La routine **5652BAND** genera a partire dal file sorgente in formato RGB 5-6-5 tre files (uno per banda) a 8bpp di estensioni RED, GRE e BLU e un descrittore di tipo DSC. La routine **CRGBCONV** compatta quindi l'immagine da 24bpp a 8bpp (cioè a 256 colori scelti con un algoritmo di ottimizzazione) utilizzando **RGBCONV** e genera un file RGB in formato TARGA tipo 1 (di estensione AFI). Quindi la routine **VGA2PMBM** converte il file AFI in un file BMP. Infine il file viene a sua volta convertito in formato RIP utilizzando la routine **PMBM2R1P**. Questo file incorpora sia il descrittore che la palette.

### *Fonte*

Centro Ricerca IBM di Milano

## AV2R1PSC

### *Sintassi*

AV2R1PSC filename

### *Ambiente*

DOS

### *Descrizione*

Questo batch file accetta il pathname completo di un file `_ID` (senza estensione) e dopo alcuni passaggi intermedi lo converte in un file formato RIP. La routine **5652BAND** genera, a partire dal file sorgente in formato RGB 5-6-5, tre files (uno per banda) a 8bpp di estensioni RED, GRE e BLU e un descrittore di tipo DSC, indispensabile per l'utilizzo di **RGBCONV2**, che compatta l'immagine da 24bpp a 8bpp (cioè a 256 colori scelti dalla palette standard `IDB$PAL$.PAL`) e genera un file raster a 1 byte per pixel che rappresenta l'indice di colore per la suddetta palette. A sua volta **TORIP** converte il raster file in un file di tipo RIP aggiungendo il relativo descrittore in coda all'immagine (ma non la palette, che può essere aggiunta successivamente con il programma **CPYRGB**).

### *Fonte*

Centro Ricerca IBM di Milano

## BJPIC

### *Sintassi*

BJPIC [= W[xH]] [opzioni] [tipo] [nome]

### *Ambiente*

AIX con X-Window

### *Descrizione*

Il programma mostra una immagine a colori su un sistema con ambiente X-Windows e OSF/Motif, utilizzando, se necessario, algoritmi di dithering o error diffusion. La sintassi di utilizzo è la stessa del package XPIC11 presente negli AIXTOOLS

Per non modificare la palette di sistema vengono utilizzati solamente 225 colori.

Gli ultimi due parametri nella lista determinano il tipo e il nome del file contenente l'immagine. Attualmente il programma riconosce i seguenti indicatori di tipo (fra i quali alcuni sono sinonimi):

.16 .565	16 bits per pixel (5 red, 6 green, 5 blu).
.4	immagine a 4 bytes per pixel: dummy, red, green e blu.
.595	immagine a colori ad 1 byte per pixel in formato RGB con 5 bits per red e blu e 9 per il green.
.DSC	file RIP con descrittore esterno. Il programma legge dal descrittore le caratteristiche del file.
.MONO .M .BW	immagine a 256 livelli di grigio 1 byte per pixel.
.R .G .B	bande red, green e blu separate su tre files ad 1 byte per pixel.
.RIP	file RIP con descrittore interno al file stesso
.RGB .3	3 bytes per pixel in un singolo file: red, green e blu.
.RLE .RT	RasterTek run-length encoded format.
.PIX .VLT	immagine ad 1 byte per pixel contenuta nel file di estensione .pix e lookup table a 3 bytes ciascuna delle 256 entry nel file di estensione .vlt

### *Fonte*

Centro Ricerca IBM di Milano

## CAPPM

### *Sintassi*

CAPPM

### *Ambiente*

OS/2

### *Descrizione*

Questo programma permette di catturare schermate grafiche in ambiente OS/2 e di stamparle o salvarle in formato BMP o PIC.

L'utilizzo del programma avviene tramite mouse e menu in ambiente Presentation Manager.

*Fonte*

CAPPM - OS2TOOLS package

## **CPY8514**

*Sintassi*

CPY8514 source.ext target.ext

*Ambiente*

DOS

*Descrizione*

Questa routine permette l'interscambio delle palette fra i formati PAL (immagine registri colore display adapter 8514/A) e RGB (formato PaintBrush + ® 769 bytes). L'estensione del source file determina il tipo di conversione. Per default si assume la conversione da PAL a RGB.

*Fonte*

Centro Ricerca IBM di Milano

## **CPYRGB**

*Sintassi*

CPYRGB source.ext target.ext < p >

*Ambiente*

DOS

*Descrizione*

Questa routine permette l'interscambio delle palette fra i formati R1P, PCX, BMP e RGB (quest'ultimo è un file di palette da 769 bytes, 1 di identificatore e 256 entries da 3 bytes R-G-B) in funzione delle estensioni utilizzate per i nomi dei files. Sono presenti numerose opzioni, fra le quali descriviamo l'opzione p (presente solo nella versione datata 12/04/1990 o successive) che considera il file in input come immagine dei registri di colore del display adapter 8514/A (1024 bytes in 256 entries R-B-G- $\Gamma$ ).

*Fonte*

Pc Paintbrush Utilities

## **CRGBCONV**

### *Sintassi*

CRGBCONV filename

### *Ambiente*

DOS

### *Descrizione*

Questa routine converte una terna di files RGB con estensioni R, G e B a un byte per pixel, più un descrittore esterno di tipo DSC, in un file AFI TARGA tipo 1 a 256 colori di nome filename.AFI.

La routine fa uso della **RGBCONV** presente nelle TGATTOOLS, che deve risiedere nella stessa directory, alla quale si rimanda per maggiori dettagli.

L'immagine ottenuta contiene una palette ottimizzata.

### *Fonte*

Centro Ricerca IBM di Milano

## **DISPPSEG**

### *Sintassi*

DISPPSEG filename

### *Ambiente*

DOS

### *Descrizione*

La routine permette di visualizzare su uno schermo VGA alcuni tipi di immagini PSEG. In particolare, non disponendo di funzioni di scaling, consente di visualizzare per intero solo i PSEG di dimensioni inferiori a quelle dello schermo. È particolarmente indicata per visualizzare il risultato di conversioni effettuate utilizzando **PIC2PSEG** o **PBIT38PP**.

### *Fonte*

PIC2PSEG package

## **FG2R1P**

### *Sintassi*

FG2R1P filename

### *Ambiente*

DOS

*Descrizione*

Questo batch file accetta il pathname completo di una terna di files di tipo RED, GRE e BLU più un descrittore di tipo DSC e, dopo alcuni passaggi intermedi, li converte in un unico file formato RIP. La routine **CRGBCONV** compatta l'immagine da 24bpp a 8bpp (cioè a 256 colori scelti con un algoritmo di ottimizzazione) richiamando la **RGBCONV**, e genera un file RGB in formato AFI TARGA tipo 1 (di estensione AFI). Quindi la routine **VGA2PMBM** converte il file AFI in un file BMP. Infine il file viene a sua volta convertito in formato RIP utilizzando la routine **PMBM2RIP** e contiene al suo interno descrittore e palette.

*Fonte*

Centro Ricerca IBM di Milano

**FG2R1PSC***Sintassi*

FG2R1PSC filename

*Ambiente*

DOS

*Descrizione*

Questo batch file accetta il pathname completo di una terna di files di tipo RED, GRE e BLU più descrittore di tipo DSC e dopo alcuni passaggi intermedi li converte in un unico file formato RIP. La routine **RGBCONV2** compatta quindi l'immagine da 24bpp a 8bpp (cioè a 256 colori scelti dalla palette standard IDB\$PAL\$.PAL) e genera un file raster a 1 byte per pixel che rappresenta l'indice di colore per la suddetta palette. A sua volta **TORIP** converte il raster file in un file di tipo RIP aggiungendo il relativo descrittore in coda all'immagine.

*Fonte*

Centro Ricerca IBM di Milano

**FRIEZE***Sintassi*

Hot-Key: Shift-PrtScr

*Ambiente*

DOS

*Descrizione*

È un programma di utilità, fornito assieme al prodotto **PaintBrush + ®**, che permette di catturare su files in formato PCX le schermate grafiche di altri programmi.

Per solo uso interno IBM

Viene installato come programma residente in memoria dallo stesso PaintBrush + ®. e può essere richiamato in qualunque momento (a patto di avere libera sufficiente memoria) premendo la combinazione di tasti **Shift-PrtScr**; compare quindi un menù che permette di salvare tutto lo schermo o una sua porzione e, eventualmente, di stamparlo.

È particolarmente indicato per realizzare presentazioni che illustrino particolari aspetti di un programma, inserendo al loro interno le relative schermate.

Il principale limite della routine consiste nel fatto che, pur essendo un programma residente, richiede che, al momento della attivazione con **Shift-PrtScr**, sia disponibile una notevole quantità di memoria non utilizzata: spesso tuttavia con immagini ad alta risoluzione non è disponibile memoria sufficiente e il sistema si blocca costringendo ad un IPL del Personal Computer.

#### *Fonte*

PaintBrush + ®

## **GIF2PB**

#### *Sintassi*

GIF2PB source.GIF target.PCX

#### *Ambiente*

DOS

#### *Descrizione*

Converte un file GIF (CompuServe Graphics Interchange Format) con 2, 4, 16, 32, 64, 128 o 256 colori in un file PCX a 256 colori.

Il file GIF, con compressione LZW, deve essere non interlacciato e contenere una sola immagine; in caso contrario viene considerata solo la prima immagine del file.

Il file PCX in output è sempre a un singolo piano; poiché alcune routines si aspettano files a più piani di bits per immagini a 16 colori, può essere necessario utilizzare in successione **PB2R1P** e **R1P2PB** per ottenere il file nel formato voluto.

#### *Fonte*

PaintBrush + ®

## **MAKEDSC**

#### *Sintassi*

MAKEDSC <-opzioni> filename

#### *Ambiente*

DOS

### Descrizione

Questa routine genera un descrittore di tipo DSC a partire dai dati passati sulla riga comandi. È comunque necessario passare a parametro, oltre al nome di file da utilizzare per il descrittore (che deve coincidere con il nome del file associato, a meno delle estensioni), le dimensioni in pixels dell'immagine.

Il nome del file deve essere l'ultimo parametro passato al programma.

### Opzioni

-?	Mostra un help con la descrizione delle opzioni disponibili
-s xx,yy	Dimensioni dell'immagine lungo x e y, in pixels. Questo parametro <b>non può essere omissso</b>
-t n	Tipo di immagine di cui scrivere il descrittore. Il parametro deve essere una stringa di caratteri scelta fra le seguenti: <b>FRAME</b> (default) immagine di tipo FRAME, cioè a look up table con palette contenuta nel file filename.PAL, a meno che un diverso palname sia specificato con l'opzione -u. <b>COLOR</b> Immagine a colori di tipo FRAME, ma con palette standard IDB\$PAL\$.PAL. <b>BW</b> Immagine a 256 livelli di grigio con palette standard BW.PAL <b>BAND</b> Immagine a colori in formato RGB, su 3 files (uno per banda) di estensioni RED, GRE e BLU.
-u palette	Nome da utilizzare per la palette. L'utilizzo di questo parametro implica una immagine di tipo FRAME.

### Fonte

Centro Ricerca IBM di Milano

## PB1T38PP

### Sintassi

PB1T38PP source.PCS dest.PP < opzioni >

### Ambiente

DOS

### Descrizione

Questo programma permette di convertire un file PCX a un singolo piano di bits (2 colori) in un file di tipo PSEG38PP stampabile in ambiente VM sulle printer delle serie 38xx.

Per default il file viene convertito in formato 120 dpi (120 punti per pollice).

Per poter essere stampato, il file deve essere trasferito su un host VM in formato binario, ad esempio con il comando:

```
mytecopy dest.PP h:dest ppbin a /b /v=256 /s
```

Occorre quindi utilizzare il programma REXX chiamato PBIN2PSEG per ultimare la conversione:

```
PBIN2PSEG dest ppbin a dest pseg38pp a
```

### *Opzioni*

- v** questa opzione specifica che ciascuna linea dell'immagine deve essere ripetuta due volte, raddoppiando così la dimensione verticale del file ottenuto
- h** specifica di utilizzare l'alta risoluzione a 240 punti per pollice. La qualità dell'immagine ottenuta è notevolmente superiore.

### *Fonte*

Pc Paintbrush Utilities

## **PB2IOCA**

### *Sintassi*

PB2IOCA source.PCX dest.IMG <nn>

### *Ambiente*

DOS

### *Descrizione*

In questo modo si esegue la conversione da formato PCX a 2, 4, 8, 16 o 256 livelli di grigio (o colori) in un file IOCA non compresso che può essere utilizzato da diversi programmi in ambiente VM.

nn è il numero di punti per pollice in X e Y che viene scritto nel descrittore del file IOCA (per default 200).

### *Fonte*

Pc Paintbrush Utilities

## **PB2R1P**

### *Sintassi*

PB2R1P sourcefn destfn

### *Ambiente*

DOS

### *Descrizione*

In questo modo si esegue la conversione da formato PCX in formato R1P. La palette contenuta nel file PCX viene inserita nel file R1P. Per inserire una palette in un file R1P che ne sia sprovvisto, è possibile utilizzare la routine **CPYRGB.EXE**.

### *Fonte*

Pc Paintbrush Utilities

## PB2TIFF

### *Sintassi*

PB2TIFF source.PCX target.TIF < opzioni >

### *Ambiente*

DOS

### *Descrizione*

Il programma converte files da formato PCX con 2, 4, 8, 16 o 256 colori/livelli di grigio a formato TIFF.

Per default il programma genera un file TIFF monocromatico o a scala di grigi compresso con l'algoritmo *PACKBITS*.

I principali TAGS vengono settati come segue:

<b>compression</b>	32773 - codice per PACKBITS
<b>planar config</b>	1
<b>samples/pixel</b>	1
<b>bits/sample</b>	n
<b>threshold</b>	1
<b>photometric interp.</b>	1 (se monocromatico)

### *Opzioni*

<b>c</b>	crea un file TIFF non compresso a 16 o 256 colori; il file in input deve essere un PCX a 16 o 256 colori.
<b>fnn</b>	crea un file monocromatico non compresso con nn punti per pollice per i tags di risoluzione in X e Y (per default 150).
<b>u</b>	genera un file monocromatico non compresso con photometric interpretation a 0.
<b>x</b>	crea un file compresso a colori di tipo PACKBITS.

### *Fonte*

Pc Paintbrush Utilities

## PIC2PSEG

### *Sintassi*

PIC2PSEG filename < opzioni >

### *Ambiente*

DOS

### *Descrizione*

Il programma mostra su uno schermo VGA una immagine nel formato PIC di StoryBoard Live e la converte in un file di tipo PSEG38PP e/o in una bitmap (senza alcun descrittore).

Per solo uso interno IBM

Il file PIC deve essere in formato a 16 colori.

Durante la fase di visualizzazione è possibile selezionare una finestra di schermo con un cursore.

Per poter essere utilizzato, il file PSEG deve essere trasferito su VM e processato con la routine **PBIN2PSE EXEC**.

La conversione da 16 a 2 colori viene effettuata utilizzando una matrice di dithering 2x2.

### *Opzioni*

?	pannello di help
/BIT = filename	l'output è una bitmap
/PSEG = filename	l'output è un file PSEG38PP
/4x4	usa una matrice di dithering 4x4
/INV	inverte i colori
/DD	utilizza la risoluzione a 120 dpi invece di 240
/ROT = 90,180,270	esegue una rotazione dell'immagine
/QUIET	non fornisce output a schermo

### *Fonte*

PIC2PSEG package

## **PMBM2R1P**

### *Sintassi*

PMBM2R1P sourcefn destfn

### *Ambiente*

DOS

### *Descrizione*

In questo modo si esegue la conversione da formato BMP in formato R1P. Il formato BMP è il formato standard utilizzato in ambiente OS/2.

### *Fonte*

Pc Paintbrush Utilities

## **PT - Picture Taker**

### *Sintassi*

Hot-Key: PrtScr

### *Ambiente*

DOS

### *Descrizione*

È un programma di utilità, fornito assieme al prodotto **IBM StoryBoard Live**, che permette di catturare su files in formato CAP le schermate grafiche di altri programmi.

Viene installato come programma residente in memoria o utilizzando il comando **PT** o dal pannello principale di StoryBoard Live e può essere richiamato in qualunque momento (a patto di avere libera sufficiente memoria) premendo il tasto **PrtScr**.

In questo modo l'immagine presente sullo schermo viene salvata e aggiunta in coda al file di tipo CAP selezionato durante l'installazione (il file **CAPTURE.CAP** per default) dal quale può essere estratta utilizzando il programma **PM - Picture Maker** e salvata in formato **PIC** o (solo per la versione Live) esportata come **PCX**, **TIFF** e altri formati.

È particolarmente indicato per realizzare presentazioni che illustrino particolari aspetti di un programma, inserendo al loro interno le relative schermate.

### *Fonte*

StoryBoard Live

## **R1P28514**

### *Sintassi*

R1P28514 - <opzioni> filename.ext

### *Ambiente*

DOS

### *Descrizione*

La routine esegue il display su video collegato alla scheda 8514/A di immagini di tipo bytemap con lookup table (formato R1P) con descrittore e palette interni o esterni. Richiede la presenza di una configurazione dual-screen (8514/A + VGA)

Normalmente la finestra di immagine selezionata viene mostrata al centro dello schermo con una cornice. L'immagine è riscalata (ridotta per fattori interi) in modo che sia possibile mostrarla per intero.

filename.ext è il file da visualizzare e deve seguire tutte le opzioni

### *Opzioni*

- ?                                   Mostra un help con la descrizione dei parametri disponibili
- a 1 0 0                           Se 1 l'immagine è posizionata con il vertice in alto a sinistra nell'origine dello schermo (o dove indicato da xx e yy se presente il parametro -w). Se 0 l'immagine viene centrata sullo schermo (default).
- c 1 0 0                           Se 0 lo schermo viene cancellato (default), se 1 l'immagine viene sovrapposta a quanto già presente.
- h 1 0 0                           Se 0 (default) il file è assunto di tipo R1P con descrittore in coda al file stesso. In caso contrario il filename.ext è supposto contenere la sola bytemap e le informazioni sul file vengono cercate nel descrittore esterno filename.DSC.

Per solo uso interno IBM

- p 1 o 0** Se 1 vengono stampate informazioni aggiuntive relativamente al file sullo schermo alfanumerico. Il caso contrario è il default.
- r 1024 o 640** Seleziona la risoluzione dello schermo grafico a 1024x768 (default) o a 640x480.
- s xx,yy,xsize,yysize** Seleziona la finestra dell'immagine, contenuta nel file, che deve essere visualizzata. (xx,yy) è il pixel in alto a sinistra, mentre xsize,yysize sono le dimensioni della finestra. Per default si visualizza l'intero file.
- u palname.ext o NULL** Indica di utilizzare la palette contenuta nel file palname.ext e di non caricare quella eventualmente indicata nel descrittore. Il programma è in grado di riconoscere palettes standard RGB (byte con valore esadecimale 0x0c seguito da 256 terne di bytes RGB che identificano le entries) o 8514 register image (1024 bytes, formati da 256 entries RBG?).
- w xx,yy,xsize,yysize** Seleziona la finestra video da utilizzare per il display dell'immagine. A meno che non sia presente il parametro -a1, xx e yy sono ignorati e l'immagine è posizionata a centro schermo. Per default si usa l'intero schermo.

### *Fonte*

Centro Ricerca IBM di Milano

## **R1P2PB**

### *Sintassi*

R1P2PB sourcefn destfn

### *Ambiente*

DOS

### *Descrizione*

In questo modo si esegue la conversione da formato R1P in formato PCX. La palette eventualmente contenuta nel file R1P viene inserita nel file PCX. Per inserire una palette in un file R1P che ne sia sprovvisto, è possibile utilizzare la routine **CPYRGB.EXE**.

### *Fonte*

Pc Paintbrush Utilities

## **R1P2PMBM**

### *Sintassi*

R1P2PMBM sourcefn destfn

### *Ambiente*

DOS

### *Descrizione*

In questo modo si esegue la conversione da formato R1P in formato BMP.

*Fonte*

Pc Paintbrush Utilities

## **R1P2R1P**

*Sintassi*

R1P2R1P <opzioni> sourcefn destfn

*Ambiente*

DOS

*Descrizione*

Permette di convertire un file in formato R1P con descrittore interno in uno con descrittore esterno e viceversa.

Per default il programma cerca il file DSC di descrizione del file sourcefn: se esiste, lo utilizza per effettuare la conversione da descrittore esterno a interno, mentre se non esiste genera un opportuno descrittore assumendo che il file sorgente abbia descrittore interno.

Il descrittore interno viene lasciato comunque in coda al file.

*Opzioni*

- e presuppone un file sorgente con descrittore interno e genera il descrittore esterno.
- i presuppone un file sorgente con descrittore esterno e inserisce il descrittore all'interno del file destinazione.
- d genera un file di destinazione eliminando il descrittore interno.

*Fonte*

Centro Ricerca IBM di Milano

## **R1P2RGB**

*Sintassi*

R1P2RGB sourcefn

*Ambiente*

DOS

*Descrizione*

Questa routine esegue la conversione da formato R1P a 256 colori (8bpp) in tre files RED GRE BLU a un byte per pixel. Il file R1P sorgente deve contenere al suo interno la palette (è possibile inserire una palette in un file R1P che ne sia sprovvisto utilizzando la routine **CPYRGB**). Viene anche creato un file DSC che contiene le informazioni relative alle dimensioni dei files.

### *Fonte*

Centro Ricerca IBM di Milano

## **R1P2VGA**

### *Sintassi*

R1P2VGA sourcefn <opzioni>

### *Ambiente*

DOS

### *Descrizione*

Consente di mostrare su un display VGA una immagine memorizzata in formato R1P o BMP. In questo ultimo caso è necessaria l'opzione M.

### *Principali opzioni*

<b>M</b>	indica che il file è di tipo BMP e non R1P
<b>R</b>	riduce l'immagine a 1/2 delle sue dimensioni e usa il modo 320*400 a 256 colori.
<b>U</b>	Inizializza la palette prelevandola dal file R1P o BMP (altrimenti viene utilizzata la palette standard della VGA)
<b>Urgb</b>	Inizializza la palette prelevandola dal file di nome rgb di tipo RGB (altrimenti viene utilizzata la palette standard della VGA)
<b>X</b>	utilizza il modo grafico 320*400 256 colori

### *Fonte*

Pc Paintbrush Utilities

## **R1PCLIP**

### *Sintassi*

R1PCLIP sourcefn.R1P destfn.R1P <x y w h> <opzioni>

### *Ambiente*

DOS

### *Descrizione*

Consente di ritagliare porzioni di immagini in formato R1P e di cambiarne le dimensioni, nonché di cambiare il numero di colori nell'immagine utilizzando tecniche di color threshold.

Il risultato della elaborazione sul file sourcefn.R1P viene scritto nel file destfn.R1P

I parametri **x** , **y** , **w** e **h** rappresentano le coordinate del vertice sinistro alto della finestra da ritagliare (**x** e **y**) e la sua ampiezza in orizzontale (**w**) e in verticale (**h**).

Vi sono numerose opzioni e pertanto si rimanda alla descrizione presente nel manuale delle Pc Paintbrush Utilities

### *Opzioni principali*

<b>D</b>	Mostra il risultato della conversione, richiamando <b>RIP2VGA</b>
<b>R</b>	Riduce l'immagine a 1/2 delle sue dimensioni (particolarmente indicata in coppia con <b>Z</b> o <b>Y</b> ).
<b>H</b>	Trasforma da 16 a 256 colori o livelli di grigio (l'opposto di <b>L</b> ) passando da 4 a 8 bpp mediante shift a sinistra di 4 posizioni.
<b>L</b>	Trasforma da 256 a 16 colori o livelli di grigio troncando i 4 bits meno significativi.
<b>P &lt;s.rgb&gt; = &lt;t.rgb&gt;</b>	Rimappa i valori dei pixel dalla palette <b>s.rgb</b> (o nulla se si vuole utilizzare la palette interna al file) sulla palette <b>t.rgb</b> . Questo risultato è ottenuto con una tecnica di errore quadratico minimo.
<b>X &lt;s.rgb&gt; &lt;= t.rgb&gt;</b>	Conversione da 256 a 16 colori utilizzando un pattern 2x2 per l'algoritmo di dithering. <b>s.rgb</b> è la palette sorgente a 256 colori, <b>t.rgb</b> è la palette destinazione a 16 colori (o la palette standard EGA se non presente).
<b>Znn</b>	Sogliatura a due colori su una finestra di 3 per 3 pixels. <b>nn</b> è il parametro di controllo della sogliatura e può assumere i seguenti valori: <b>0</b> evidenzia le aree nere; <b>1...9</b> numero di pixels neri nella finestra per ottenere un pixel nero; <b>11...19</b> come sopra, ma mette il pixel a nero solamente se lo è già; <b>10</b> nero se parte di una sequenza orizzontale (run) di pixels neri;
<b>Ynn</b>	Sogliatura a 16 colori su una finestra di 3 per 3 pixels. <b>nn</b> è il parametro di controllo della sogliatura e può assumere i seguenti valori: <b>0</b> evidenzia le aree colorate; <b>1...9</b> numero di pixels dello stesso colore nella finestra per forzare il pixel a quel colore; <b>11...19</b> come sopra, ma usa il bianco per valori sotto il livello di soglia; <b>21...29</b> come sopra, ma usa il colore più frequente nella finestra se sotto il livello di soglia; <b>10</b> colore più frequente nella finestra; <b>20</b> secondo colore più frequente nella finestra; <b>30</b> evidenzia in nero i contorni e svuota le aree colorate; <b>31</b> come sopra, ma lascia le aree colorate;

### *Fonte*

Pc Paintbrush Utilities

## **RGBCONV**

### *Sintassi*

RGBCONV - <opzioni> filename

### *Ambiente*

DOS

### *Descrizione*

Per solo uso interno IBM

Questa routine converte una terna di files RGB con estensioni R, G e B a un byte per pixel in un file AFI TARGA tipo 1 a colori di nome filename.AFI o di tipo VGA con nome filename.VGA.

Viene effettuata una ottimizzazione della palette, senza però utilizzare tecniche di dithering o error diffusion.

Quando viene richiesto un output di tipo VGA, l'immagine viene riscalata alla dimensione di 360x480, o quanto più possibile vicina a questa, ma senza superarla, in modo da mantenere il rapporto d'aspetto (aspect ratio) dell'immagine originale.

Il nome della terna di files da convertire (filename **senza estensione**) deve essere posto dopo tutte le opzioni, e può contenere un pathname.

Tutti i parametri possono essere specificati sulla linea comandi, o inseriti in modo interattivo dal pannello di dialogo generato dal programma.

### *Opzioni*

-c nn	numero di colori desiderati per l'output (1..256, default 256)
-w nn	ampiezza in pixels dell'immagine
-h nn	altezza in pixels dell'immagine
-rn	numero di bits del rosso da utilizzare (5 per default)
-gn	numero di bits del verde da utilizzare (5 per default)
-bn	numero di bits del blu da utilizzare (5 per default)
-v	genera un file di tipo VGA (per default è AFI)
-Gx.x	usa un fattore gamma per la conversione x.x (il default è 1.0)
-Lx.x	usa una correzione x.x per la luminosità (il default è 1.0)

### *Fonte*

TGATOOLS

## **RGBCONV2**

### *Sintassi*

RGBCONV2 source dest

### *Ambiente*

DOS

### *Descrizione*

Questa routine converte una terna di files RGB (con estensioni RED, GRE e BLU a un byte per pixel) più un descrittore di tipo DSC in una immagine in formato RIP con descrittore esterno sempre di tipo DSC.

I nomi per i files di partenza -source- e destinazione -dest- devono essere **privi di estensione**, ma possono contenere un pathname.

L'immagine viene convertita portandola da 24 bpp a 256 colori appartenenti alla palette standard IDB\$PAL\$.PAL, che è la palette indicata nel descrittore.

Viene anche generata una copia della palette standard con il nome dest.PAL

*Fonte*

Centro Ricerca IBM di Milano

## **SHI6090**

*Sintassi*

SHI6090 fname fmode

*Ambiente*

VM/CMS

*Descrizione*

Mostra su 6090 in ambiente VM una immagine memorizzata in 3 files (uno per banda) di filename *fname* e di estensioni RED GRE e BLU, a un byte per pixel per banda.

La lunghezza del record rappresenta il numero di pixels per linea; il numero di records rappresenta invece il numero di linee nell'immagine. Pertanto, per trasferire da PS/2 su VM una terna di files *xx* per *yy* occorre dare i comandi (utilizzando ad esempio MYTE):

```
mytecopy fname.red h:fname red a /lrecl=xx /b
mytecopy fname.gre h:fname gre a /lrecl=xx /b
mytecopy fname.blue h:fname blue a /lrecl=xx /b
```

L'immagine mostrata su 6090 può essere ritoccata nei colori agendo sui livelli medi di rosso, verde e blu della palette e quindi stampata con la hard copier **IBM 6097 Screen Printer**

*Fonte*

Centro Ricerca IBM di Milano

## **TIFF2PB**

*Sintassi*

TIFF2PB source.TIF target.PCX <g>

*Ambiente*

DOS

*Descrizione*

Il programma converte files da formato TIFF a formato PCX.

Il formato TIFF comprende numerose varianti e diverse possibilità per gli algoritmi di compressione utilizzati: la routine riconosce files monocromatici non compressi o compressi utilizzando l'algoritmo *PACKBITS* o la compressione CCITT gruppo 3 "modified Huffman run length".

I dati devono essere memorizzati in una unica strip e possono essere sia in formato INTEL che MOTOROLA.

Per solo uso interno IBM

Se l'immagine è a livelli di grigio (2, 4 0 8 bits per pixel) il file generato è un PCX a 4, 16 o 256 colori.

### *Opzioni*

**g** viene inserita nel file una palette a scala di grigi.

### *Fonte*

Pc Paintbrush Utilities

## **TOR1P**

### *Sintassi*

TOR1P source.RIP = xx yy bpl bpp <d>

TOR1P source.KPS dest.RIP k <i> <d>

TOR1P source.PIX dest.RIP p <d>

TOR1P source.PIX dest.RIP x <m palette.RGB>

TOR1P source.xxx dest.RIP <xx <yy>> y <m palette.RGB>

TOR1P source.xxx dest.RIP xx yy bpl bpp <offset> <i> <d>

### *Ambiente*

DOS

### *Descrizione*

La routine consente di estrarre una immagine in formato RIP con descrittore interno a partire da numerosi tipi di files contenenti l'immagine sotto forma di un singolo piano raster non compresso.

Il programma può essere eseguito utilizzando una delle forme sopra descritte, in funzione del tipo di file sorgente. La prima forma è utilizzabile quando il file è già in formato RIP, ma con descrittore interno. In questo caso **non viene creato un nuovo file, ma viene modificato quello originale.**

I parametri non racchiusi fra i segni '<' e '>' non possono essere omessi.

### *Opzioni*

<b>xx</b>	larghezza in pixels dell'immagine
<b>yy</b>	altezza in pixels dell'immagine
<b>bpl</b>	numero di bytes per ciascuna linea dell'immagine raster
<b>bpp</b>	numero di bits per ciascun pixel
<b>offset</b>	offset dell'immagine nel file, cioè posizione del byte in cui inizia l'immagine raster
<b>k</b>	si assume che l'immagine di partenza sia di tipo <b>KPS</b> ; in questo caso i parametri relativi vengono letti dall'header del file stesso.
<b>p</b>	si assume che l'immagine in ingresso sia di tipo <b>PIX</b> con dimensioni 256x240 a 3 bytes RGB per pixel con range da 0 a 63; l'immagine viene in questo caso portata a 16 colori raddoppiandone le dimensioni e applicando una tecnica di dithering. Viene utilizzata la palette standard a 16 colori della scheda EGA.
<b>x</b>	analogo al parametro p, ma l'immagine prodotta è a 256 colori con la palette standard di PaintBrush + ®

**y** analogo a x ma con range fra 0 e 255 invece che fra 0 e 63  
**m palette.rgb** permette, nel caso in cui si utilizzi il parametro x o y, di utilizzare una palette diversa da quella standard  
**i** inverte l'immagine

*Fonte*

Pc Paintbrush Utilities

## **VGA2PMBM**

*Sintassi*

VGA2PMBM source.ext dest <.BMP> <f>

*Ambiente*

DOS

*Descrizione*

Il programma converte un file a 256 colori di tipo AFI targa 1 e di nome source.AFI o di tipo VGA con nome source.VGA in un file standard OS/2 Presentation Manager a 256 colori di nome dest.BMP.

Se viene omessa l'estensione per il file di destinazione, il programma genera un file di tipo R1P con nome dest.R1P e una palette chiamata dest.RGB.

L'opzione f inverte verticalmente l'immagine.

*Fonte*

Pc Paintbrush Utilities





## Appendice B. I formati più diffusi

### Formati raster

<b>3</b>	Immagini RGB a 3 bytes per pixel (red, green, blue) in un singolo file
<b>4</b>	Immagini RGB a 4 bytes per pixel (dummy, red, green e blue) in un singolo file
<b>16</b>	file RGB a 16 bits per pixel (5 red, 6 green ,5 blu)
<b>565</b>	file RGB a 16 bits per pixel (5 red, 6 green ,5 blu)
<b>595</b>	file utilizzato da xpict1 e derivati (fra i quali bjpict) per memorizzare immagini error diffused
<b>ART</b>	PFS First Publisher cutout
<b>AFI</b>	PCTOOLS TGATOOLS TARGA 1
<b>BIT</b>	Lotus Manuscript
<b>BMP</b>	OS/2 PM bitmap, MS Windows DIB bitmap
<b>BSV</b>	Microsoft Basic BSAVE of CGA, EGA, VGA frame buffer
<b>CAP</b>	Storyboard Live Picture Taker files
<b>CLP</b>	MS Windows Clipboard
<b>CUT</b>	Media Cybernetics Dr. HALO
<b>DIB</b>	Windows V3 bitmap
<b>DSC</b>	Descrittore per immagini RIP su file esterno
<b>GIF</b>	Graphics Image File, standard utilizzato da CompuServe
<b>GXI</b>	Show Partner
<b>IAX</b>	Grafica raster su host IBM
<b>IFF</b>	Electronic Arts Image File Format (Commodore Amiga)
<b>IMG</b>	IBM IOCA o GEM image
<b>KPS/KPL</b>	IBM 8514 raster e palette formato in memoria
<b>KPS/PAL</b>	IBM 8514 raster e palette formato in memoria (MAND8)
<b>LBM</b>	Deluxe Paint II (identico a IFF)
<b>MAC</b>	MacPaint, PFS First Publisher
<b>MSP</b>	Microsoft Windows Paint (antecedente a Windows V3)
<b>PCC</b>	ZSoft PcPaintbrush clipboard (antecedente a ver. IV)
<b>PCX</b>	ZSoft PcPaintbrush
<b>PF</b>	PFS First Choice/Professional write (format di stampa)
<b>PIC</b>	Storyboard Plus, o HALO, o PC Paint + , o Pictor, o Grasp, o IBM MLS(LS/1), o .... (vedere anche formati vector)
<b>PIX</b>	InSet, o Vision technologies
<b>PSEG</b>	IBM Advanced Function Print Stream image segment
<b>PNT</b>	FullPaint (Macintosh)
<b>RIP</b>	PCTOOLS PBRUSHUT raster a colori
<b>RAS</b>	IBM PCIDU raster file
<b>RED</b>	Assieme a GRE e BLU immagine raster a tre bande su files separati
<b>R</b>	Assieme a G e B immagine raster a tre bande su files separati
<b>RLE</b>	CompuServe "Run length Encoded" o Windows V3 DIB compresso
<b>RL4</b>	Windows V3 DIB compresso
<b>RL8</b>	Windows V3 DIB compresso
<b>RGB</b>	Immagini RGB a 3 bytes per pixel (red, green e blue) in un singolo file
<b>SCX</b>	RIX - EGA Paint, ecc.
<b>TGA</b>	Formati immagine TARGA
<b>TIF</b>	TIFF (Microsoft/Aldus - both PC and MAC)
<b>WPG</b>	Word Perfect
<b>VGA</b>	PCTOOLS TGATOOLS TARGA tipo 1
<b>VID</b>	IBM MMotion Image
<b>_ID</b>	IBM AVC RGBImage format

### Formati vector

<b>CGM</b>	Computer Graphics Metafile - Lotus &FL., Harvard Graphics, Zenographics' Pixie, e Aldus PageMaker
<b>DRW</b>	Lotus Freelance or Micrografx software products
<b>DXB</b>	Versione binaria (compilata) di DXF
<b>DXF</b>	AutoDesk AutoCAD
<b>EPS</b>	Encapsulated PostScript

<b>GEM</b>	Formato GEM Object-oriented
<b>GL</b>	(HPGL/IBMGL) Hewlett-Packard Graphics Language per plotters
<b>GMF</b>	Come CGM
<b>MET</b>	OS/2 PM Metafile
<b>PCL</b>	Hewlett-Packard Printer Control Language (Laserjet)
<b>PIC</b>	Grafici Lotus 1-2-3
<b>PIF</b>	IBM Picture Interchange Format
<b>PICT</b>	MacDraw, Macintosh software.
<b>SLD</b>	Formato slide AutoCAD

## Appendice C. Files utilizzati

Tabella 1 (Pagina 1 di 2). Files utilizzati			
<i>File</i>	<i>OS</i>	<i>Descrizione</i>	<i>Fonte</i>
<b>PROGRAMMI DI CONVERSIONE</b>			
5652BAND	DOS	565 o _ID -> R1P	RSTMI
AV2R1P	DOS	_ID -> R1P palette ottimizzata	RSTMI
AV2R1PSC	DOS	_ID -> R1P palette standard	RSTMI
CAPPM	OS/2	Cattura schermate OS/2 e salva in formato BMP	PCTOOLS
CPY8514	DOS	PAL 8514 <-> RGB	RSTMI
CPYRGB	DOS	estrazione e conversione palette	PBUTI
CRGBCONV	DOS	R, G, B, DSC -> AFI 256 colori palette ottimizzata	RSTMI
FG2R1P	DOS	RED GRE BLU -> R1P palette ottimizzata	RSTMI
FG2R1PSC	DOS	RED GRE BLU -> R1P palette standard	RSTMI
FRIEZE	DOS	Cattura schermate in formato PCX	PBRUSH
GIF2PB	DOS	GIF -> PCX	PBUTI
MAKEDSC	DOS	Generazione descrittore esterno DSC	RSTMI
PBIT38PP	DOS	PCX -> PSEG38PP (necessita di PBIN2PSE EXEC su VM)	PBUTI
PB2IOCA	DOS	PCX -> IOCA	PBUTI
PB2R1P	DOS	PCX -> R1P	PBUTI
PB2TIFF	DOS	PCX -> TIF	PBUTI
PBIN2PSE	VM	da utilizzare assieme a PBIT38PP per la parte VM	PBUTI
PIC2PSEG	DOS	converte files di tipo PIC in files PSEG binari	STB
PMBM2R1P	DOS	BMP -> R1P	PBUTI
PT	DOS	Picture Taker - cattura schermate in formato PIC	STB
R1P2PB	DOS	R1P -> PCX	PBUTI
R1P2PMBM	DOS	R1P -> BMP	PBUTI
R1P2R1P	DOS	R1P int desc <-> R1P ext desc	RSTMI
R1P2RGB	DOS	R1P -> RED GRE BLU DSC	RSTMI
RGBCONV	DOS	RED GRE BLU -> AFI palette ottimizzata	TGA
RGBCONV2	DOS	RED GRE BLU -> R1P standard palette	RSTRM
TIFF2PB	DOS	TIFF -> PCX	PBUTI
TOR1P	DOS	bytemap -> R1P	PBUTI
VGA2PMBM	DOS	AFI -> R1P o BMP	PBUTI
<b>PROGRAMMI DI ELABORAZIONE</b>			
R1PCLIP	DOS	cut, resize e cambio colori in immagini R1P	PBUTI
<b>PROGRAMMI DI VISUALIZZAZIONE</b>			
BJPIC	AIX	Mostra files di diversi formati in ambiente X-Windows	RSTMI
DISPPSEG	DOS	Mostra su VGA files PSEG binari	PIC2PSEG

<b>Tabella 1 (Pagina 2 di 2). Files utilizzati</b>			
<i>File</i>	<i>OS</i>	<i>Descrizione</i>	<i>Fonte</i>
<b>RIP28514</b>	DOS	RIP su display adapter 8514/A	<b>RSTMI</b>
<b>RIP2VGA</b>	DOS	RIP o BMP su scheda VGA	<b>PBUTI</b>
<b>SHI6090</b>	VM	RED GRE BLU su IBM6090 in ambiente VM	<b>RSTMI</b>
<b>PROGRAMMI DI ACQUISIZIONE</b>			
<b>ICONA</b>	DOS	Acquisizione con scanner per diapositive Eikonix	<b>RSTMI</b>
<b>TVCAMER</b>	DOS	Acquisizione con telecamera da PC4000-FG4000	<b>RSTRM</b>
Nota: La fonte è codificata come segue			
<b>PBRUSH</b>	PaintBrush+ ®		
<b>PBUTI</b>	PCPAINTBRUSH Utilities - PcTools		
<b>PCTOOLS</b>	PcTools package		
<b>PIC2PSEG</b>	PIC2PSEG package - PcTools		
<b>RSTMI</b>	Centro Ricerca IBM di Milano		
<b>RSTRM</b>	Centro Ricerca IBM di Roma		
<b>STB</b>	StoryBoard Live		
<b>TGA</b>	TGATOOLS - PcTools		

## Appendice D. I programmi sul dischetto allegato

Tabella 2. Routines presenti sul dischetto allegato			
<i>File</i>	<i>OS</i>	<i>Descrizione</i>	<i>Source</i>
5652BAND	DOS	565 o _ID -> RIP	SI
AV2RIP	DOS	_ID -> RIP palette ottimizzata	SI
AV2RIPSC	DOS	_ID -> RIP palette standard	SI
BJPIC	AIX	Mostra files di diversi formati in ambiente X-Windows	NO
CPY8514	DOS	PAL 8514 <-> RGB	SI
CPYRGB	DOS	estrazione e conversione palette	SI
CRGBCONV	DOS	R, G, B, DSC -> AFI 256 colori, palette ottimizzata	SI
FG2RIP	DOS	RED GRE BLU -> RIP palette ottimizzata	SI
FG2RIPSC	DOS	RED GRE BLU -> RIP palette standard	SI
MAKEDSC	DOS	Generazione descrittore esterno DSC	SI
RIP28514	DOS	RIP su display adapter 8514/A	NO
RIP2RIP	DOS	RIP int desc <-> RIP ext desc	SI
RIP2RGB	DOS	RIP -> R G B DSC	SI
RGBCONV	DOS	RED GRE BLU -> AFI optimized palette	NO
RGBCONV2	DOS	RED GRE BLU -> RIP standard palette	NO
SIH6090	VM	RED GRE BLU su IBM6090 in ambiente VM	NO
Nota: Per utilizzare i batch files, è necessario disporre anche di tutti i files richiamati (vedere Appendice C per le fonti)			



## Appendice E. PBRUSHUT: i programmi del package

ADJIMGR	Visualizza IOCA header (files IMG) e consente modifiche di risoluzione.
ART2PB	Da 1st Publisher o 1st Choice a PCC
BMP2PB	Da OS/2 PM BMP a PCX
BYT2PB	Da immagine raster a 8 bits a PCX 256 colori
CPYRGB	Trasferisce palette RGB fra files BMP, PCX, RGB, e RIP
DIB2PMBM	Da Windows Version 3 DIB (BMP) a OS/2 PM BMP
EADUMP	Manda il contenuto di un file sullo standard output
EPAL2RGB	Da palette EGA a palette RGB
FCAD2PB	Da Generic CAD raster a PCX
GBYTE2PB	Da immagini raster 8 bit, scala di grigi, a PCX 16 livelli di grigio
GIF2PB	Da Compuserve GIF non interallacciato a PCX
IIF2PB	Da Electronics Arts IFF 85 a PCX
LS2PB	Da Logitech ScanMan a PCX bianco e nero o scala di grigi
MAC2PB	Da PFS ART a PCC
MIXRGB	Miscela 2 palettes RGB a 256 colori
MSP2PB	Da Windows MSP a PCX
PB14T12	Da PCX 1 piano 4 colori a 1 piano 2 colori
PB1T38PP	Da PCX a PSEG38PP
PB256T16	Da PCX 256 colori a 16 colori con dithering
PB2BMP	Da PCX a BMP customizzato
PB2CGA	Visualizza su CGA immagini PCX a 2 e 4 colori
PB2EGA	Visualizza immagini PCX su video EGA (emulazione 256 colori)
PB2IOCA	Da PCX a IMG (IBM IOCA)
PB2PFH	Da PCX a PFS PF alta risoluzione o graphics printer
PB2PFS	Da PCX a PFS PF bassa risoluzione o graphics printer
PB2R1P	Da PCX a R1P
PB2RAS	Da PCX bianco e nero a PCIDU RAS
PB2TIFF	Da PCX scala di grigi o colori a TIFF
PB2VGA	Visualizza immagini PCX su una VGA
PBAPAL	Modifica palettes a 256 colori per immagini PCX (richiede VGA)
PBCTMON	Da PCX a colori a PCX bianco e nero
PBMONTC	Da PCX bianco e nero a PCX a colori
PBMPAL	Rimappa immagini PCX a colori su una nuova palette
PEBO2PB	Cattura schermate PEEKABO in PCX
PIC256PB	Da Storyboard PIC 320x200 256 colori a PCX
PL2PBPIC	Da IBM/HP GL a LOTUS PIC
PMBM2R1P	Da OS2 PM BMP, icona o puntatore to R1P
RIP2EGA	Visualizza RIP, BMP or RAS su EGA (emulazione 256 colori)
RIP2PB	Da R1P a PCX
RIP2PMBM	Da R1P a BMP
RIP2VGA	Visualizza R1P, BMP, o raster su una VGA
RIPCLIP	Elaborazione: "clip", "enhance", "convert color range", etc. su R1P
RIPZOOM	"Shrink"/"grow" e/o "rotate" su R1P
RAS2PB	Da PCIDU RAS a PCX
TIFF2PB	Da TIFF a colori o scala di grigio a PCX
TOR1P	Da KPS, PIX e raster generico a R1P
VGA2PMBM	Da Targa 1 (AFI/VGA) a BMP o R1P



## Appendice F. Utili prodotti software e TOOLS

Tabella 3. Descrizione di alcuni utili prodotti software e TOOLS	
Prodotto	Descrizione
Audio Visual Connection IBM	Programma per la realizzazione di presentazioni grafiche in ambiente DOS e OS/2. È dotato di un ottimo ambiente per l'acquisizione di immagini attraverso la scheda Video Capture Adapter e per il post-editing delle stesse immagini (che hanno una risoluzione di 16 bits per pixel in formato 5 red, 6 green, 5 blu).
CAPPM V.R. Zayas, IBM OS2TOOLS package	Package in ambiente OS/2 per la cattura di schermate o finestre di applicazioni. Le finestre catturate possono essere stampate o salvate su file in diversi formati, fra cui BMP e PIC.
OS/2 Image Support IBM	Prodotto in ambiente OS/2 per la visualizzazione e la conversione di immagini in diversi formati, fra i quali BMP, PCX e TIFF.
Pc Paintbrush Utilities (PBRUSHUT) John Hind, IBM PCTOOLS package	Ottimo package disponibile sui PCTOOLS IBM e costituito da numerose routines di conversione di immagini in ambiente DOS.
PaintBrush + ® ZSoft Corporation	Package di editing di immagini in ambiente DOS. Il suo formato di immagini (PCX) è lo standard più diffuso nel mondo dei Personal Computer per le immagini a colori. Al suo interno comprende la routine FRIEZE molto utile per la cattura di schermate di altri programmi.  L'ambiente operativo Microsoft Windows 3.0 ne fornisce come accessorio una apposita versione.
PIC2PSEG Jürg von Känel, IBM PCTOOLS package	Package in ambiente DOS per la conversione di files da formato PIC o PCX a PSEG. Il package contiene alcune utili routine di conversione e un programma per la visualizzazione su Personal Computer di immagini in formato PSEG di piccole dimensioni. Il manuale contiene una utile descrizione dei formati PIC e PCX.
PS2PSEG Terry Foster, IBM TXTTOOLS package	Package in ambiente VM per la conversione di files da formato Encapsulated PostScript a PSEG. Il package richiede l'IBM Publishing System - PostScript Interpreter.
Storyboard Live IBM	Diffusissimo prodotto per la realizzazione di presentazioni grafiche in ambiente DOS. È costituito da 4 programmi principali :  <b>PICTURE MAKER</b> Programma di editing di immagini molto semplice ma efficace. Nella versione Live del prodotto, diversamente dalla precedente versione Plus, è presente una funzione che permette l'esportazione delle immagini in diversi formati, fra cui PCX e TIFF.  <b>PICTURE TAKER</b> Utilizzato per la cattura di schermate grafiche di altri programmi e l'importazione in presentazioni.  <b>STORY EDITOR</b> Authoring system per la realizzazione delle presentazioni  <b>STORY TELLER</b> Modulo run-time per la gestione delle presentazioni realizzate.
TGATOOLS Alberto DiBene, IBM PCTOOLS package	Insieme di routines di conversione in ambiente DOS per immagini in formato TARGA



## Bibliografia

- [1] D. Pomerantz - **A Few Good Colors** - Computer Language, August 1990
- [2] **Audio Visual Connection, user's guide** - IBM, 1990
- [3] S. Rimmer - **Bit Mapped Graphics** - Windcrest Books, 1990
- [4] W. Pratt - **Digital Image Processing** - J. Wiley & Sons, 1978
- [5] G. Goertzel, G.R. Thompson - **Displaying Scanned Images with Faithful Color on an 8514** - IBM Research Report, RC 15038 (#67060) 10/05/1989
- [6] A. Meadow, R. Offner M. Budiansky - **Handling Image Files with TIFF** - Dr. Dobb's Journal, May 1988
- [7] R. Hunter and A.H. Robinson - **International Digital Facsimile Coding Standards** - Proceedings of the IEEE, 68,7 1980
- [8] S. Apiki - **Lossless Data Compression** - Byte, March 1991
- [9] J. Hind - **Pc Paintbrush Utilities (PBRUSHUT)** - IBM PCTOOLS package, 1991
- [10] **PaintBrush+ ®, user's manual** - ZSoft Corporation, 1987
- [11] J. von Kanel - **Pic2Pseg Package** - IBM PCTOOLS package, 1989
- [12] **Storyboard Live, user's guide** - IBM, 1990
- [13] **Tag Image File Format - Rev 4.0** - An Aldus/Microsoft Technical Memorandum, 1987
- [14] A. DiBene - **TGATOOLS** - IBM PCTOOLS package, 1990
- [15] F. Montaniz - **Techniques for Capturing, Displaying and Printing PC Screens** - IBM Research Report, RC 15776 (#70158) 05/21/1990
- [16] K. Quirk - **Translating PCX Files** - Dr. Dobb's Journal, August 1989

**IBM** IBM Semea  
Milan Scientific Center  
Circonvallazione Idroscalo  
I-20090 Segrate (Milano, Italy)