

Elaborazione di Segnali e Immagini (ESI)

AA 2003-2004

Paola Bonetto

email: bonetto@disi.unige.it

Stanza: #110

Tel: 010 353 6643

Il formato PCX

- ❑ Nato per PC Paintbrush della Z-Soft
- ❑ Supporta 1-, 4-, 8-, 16-, 256-color palette e 24-bit true color: l'evoluzione del formato è al passo dei progressi tecnologici
- ❑ Ottima compressione (RLE): per questo è ancora molto diffuso, per es. nell'industria dei video-giochi
- ❑ Ideato per piattaforma PC: formato little-endian

La struttura del file PCX

- Header
 - Signature
 - Info immagine
 - Palette 16 colori RGB
- Immagine compressa
- Palette di colori RGB per la decodifica di immagini a 256 colori (opzionale)

Elaborazione Segnali e Immagini

3

Lo Header PCX

| Campo | Dimensione | Valore | Descrizione |
|-------------------|--------------------|---------|------------------------------|
| Identifier | Byte | \$10 | PCX ID |
| Version | Byte | 0, 2-5 | Versione PCX Paintbrush |
| Encoding | Byte | 1 | 0-non codific. 1-codificato |
| BitsPerPixel | Byte | 1-8 | Bits x piano colore per pxl |
| Xstart, Ystart | Word, Word | 0, 0 | Angolo sup sx immagine |
| Xend, Yend | Word, Word | *, * | Angolo inf dx immagine |
| HorzRes, VertRes | Word, Word | *, * | Risoluzione scanning |
| Palette | Array[0..47], byte | * | Array 48-byte di colori RGB |
| Reserved1 | Byte | 0 | Riservato per uso futuro |
| NumBitPlanes | Byte | 1-4 | # piani colore |
| BytesPerLine | Word | * | Bytes per riga dell'img |
| PaletteType | Word | 0, 1, 2 | Ignora colr/mono scala grigi |
| Horz-,VertScnSize | Word, Word | *, * | Dim schermo (ignorabile) |
| Reserved2 | Array[0..53], byte | 0s | Padding |

La palette RGB

| Campo | Dimensione | Descrizione |
|------------|-----------------------------|-----------------|
| Signature | Byte | Sempre \$0C hex |
| Palette256 | Array[0..255, 0..2] di Byte | Triple RGB |

Elaborazione Segnali e Immagini

5

Il formato BMP

- Formato bitmapped standard per piattaforme Windows
- Un file BMP è organizzato in 2 o 3 blocchi:
 - Header che consiste di due sottoparti:
 - BitmapFileHeader e
 - BitmapCoreHeader (per OS/2 o vs <Win3.0) o BitmapInfoHeader (per tutte le altre vs di Windows)
 - Palette (opzionale)
 - Organizzata in triple o quadruple a seconda della versione dell'OS
 - Dati dell'immagine

Elaborazione Segnali e Immagini

6

BitmapFileHeader

| Campo | Dimensione | Descrizione |
|--------------|-------------------|---|
| Tipo | Word | Caratteri "B" e "M" |
| Dimensione | LongInt | Dimensione file in bytes |
| Riservato1 | Word | Riservato Microsoft |
| Riservato2 | Word | Riservato Microsoft |
| Offset | LongInt | Offset in bytes per l'inizio del bitmap |

BitmapCoreHeader

| Campo | Dim. | Descrizione |
|---------------|-------------|------------------------------|
| Dimensione | LongInt | Dimensione di questo header |
| Larghezza | Integer | Larghezza del BMP in pixel |
| Altezza | Integer | Altezza del BMP in pixel |
| Piani | Word | # piani di colore (sempre 1) |
| ConteggioBits | Word | Bits per pixel |

BitmapInfoHeader

| Campo | Dim. | Descrizione |
|------------------|-------------|-------------------------------|
| Dimensione | LongInt | Dimensione di questo header |
| Larghezza | LongInt | Larghezza del BMP in pixel |
| Altezza | LongInt | Altezza del BMP in pixel |
| Piani | Word | # piani di colore (sempre 1) |
| ConteggioBits | Word | Bits per pixel |
| Compressione | LongInt | Flag di compressione |
| DimensioneImg | LongInt | Dimensione in bytes dell'img |
| XRes,Yres | LongInt x 2 | Risoluzione: Pixel al metro |
| ColoriUsati | LongInt | # colori nella tavola di clr. |
| ColoriImportanti | LongInt | # colori importanti |

La palette RGB (tripla o quad.)

| Campo | Dimensione | Descrizione |
|--------------|-------------------|-----------------------------------|
| Blue | Byte | Intensità blu 0..255 |
| Green | Byte | Intensità verde 0..255 |
| Red | Byte | Intensità rosso 0..255 |
| Riservato | Byte | Flags (solo nel nuovo formato) |

I dati dell'immagine

Numerazione dei pixels dell'immagine:

```
0 1 2 3 4 5 6 7 8.....54 55 56 57 58 59
```

Righe nel file BMP (img a 256 colori):

```
63 01 24 A4 39 7F BC 57 00 83.....2E 89 D0 00 00 00
```

```
62 A2 B3 31 49 F7 CB 59 03 84.....F1 65 0F 00 00 00
```

```
...
```

```
0 54 CA E2 74 82 05 B3 21 CF.....AA 47 92 00 00 00
```

Elaborazione Segnali e Immagini

11

Il formato TGA (Targa)

- Sviluppato per i display Targa
- Divenuto noto con la diffusione di scanner
- Upgraded per il supporto di immagini a colori con 15-, 16-, 24- e 32-bits-per-pixel
- "Uncompressed bitmap format":
compressione supportata ma non diffusa (RLE)
- Legge dall'alto al basso o viceversa;
da sinistra a destra o da destra a sinistra
- Sviluppato per piattaforma PC: little endian

Elaborazione Segnali e Immagini

12

Struttura del file TGA

- Versione 1.0:
 - Header con informazioni su immagine e palette
 - Un campo con ID dell'immagine (opzionale)
 - Mappa di colore (opzionale)
 - Dati bitmap
- Vs. 2.0: seguono altri campi, opzionali., tra cui x es:
 - Info sviluppatore, tabella di correzione colore, img thumbnail (*postage-stamp image*), footer.

Elaborazione Segnali e Immagini

13

Il formato TIFF (Tagged Image File Format)

- Aldus & Microsoft
- Scopo: trattare il più alto nr possibile di "tipi" di img; es: B&W, compresse con PackBits, etc.
- < Vs 6.0: classi TIFF di immagini; per es:
 - Classe B: B&W
 - Classe R: RGB
 - Classe P: immagini indicizzate o basate su palette
 - Classe G: scala di grigio
- Basato su Tags: pbl di complessità, incompatibilità =>
- Vs. 6.0: TIFF standard

Elaborazione Segnali e Immagini

14

TIFF baseline: es. campi obbligatori

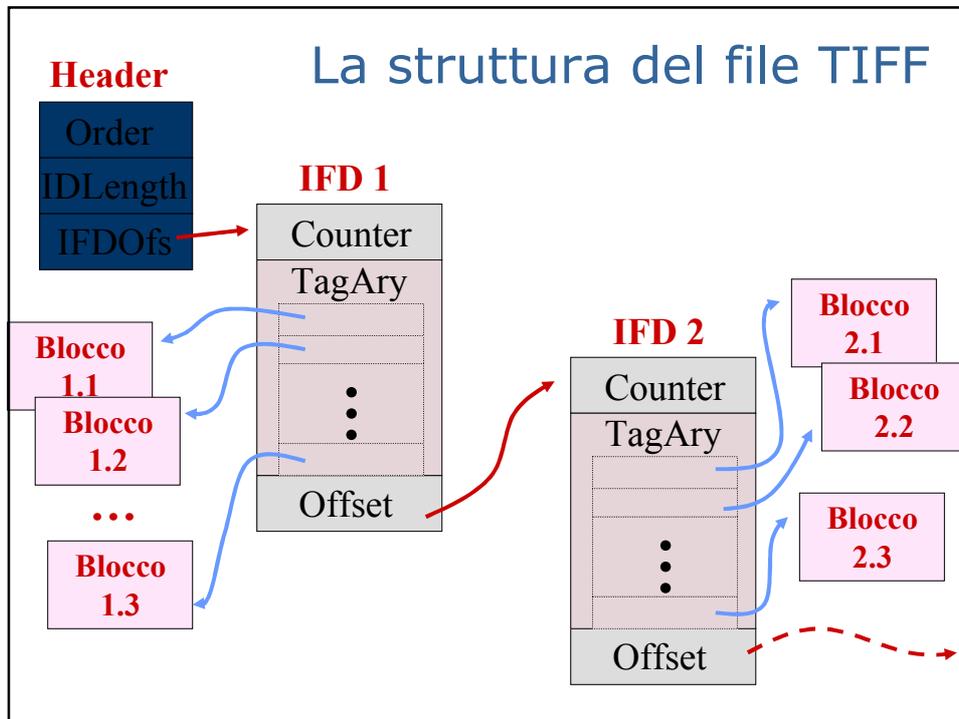
| Campo | Tag | Dflt | B/W | Gray | Pltte | RGB |
|-------------|----------|------|-----|------|-------|-----|
| ImgWidth | \$0100 h | | √ | √ | √ | √ |
| ImgHeight | \$0101 h | | √ | √ | √ | √ |
| ColorMap | \$0140 h | | | | √ | |
| Compression | \$0103 h | 1 | √ | √ | √ | √ |
| XResolution | \$011A h | | √ | √ | √ | √ |
| YResolution | \$011B h | | √ | √ | √ | √ |
| ResUnit | \$0128 h | 2 | √ | √ | √ | √ |

Elaborazione Segnali e Immagini

15

TIFF baseline: es. campi facoltativi

| Campo | Tag | Tipo | Conteggio | Valore |
|----------------|----------|------|------------------------|------------------------------|
| Artista | \$013B h | Char | Illimitato | Ptr o nome ASCII |
| Copyright | \$8298 h | Char | # Chrs + 1 | Stringa+NULL |
| DateTime | \$0132 h | Char | 20 | Ptr a stringa ASCII |
| FillOrder | \$010A h | Word | 1 | 1 o 2 |
| GrayRspnsCrve | \$0123 h | Word | $2^{\text{Bits/Smpl}}$ | Ptr a tabella |
| HostComputer | \$013C h | Char | Unlimited | Nome o ptr al nome |
| ImgDescription | \$010E h | Char | Unlimited | Descr.o ptr a descr. |
| MaxSampleVal. | \$0119 h | Word | SmpIs/Pxl | $(2^{\text{Bits/Smpl}}) - 1$ |
| Orientation | \$0112 h | Word | 1 | 1 to 7 |
| Software | \$0131 h | Char | Unlimited | Nome Sw o ptr |



La struttura del file TIFF

| Header | | |
|------------|--------------------|--|
| Campo | Dimensione | Descrizione o valore |
| TIFFOrder | Array[0,1] di Char | "II"=little endian, "MM"=big endian |
| TIFFIDLen | Word | \$0042 hex (nr.versione) |
| TIFFIFDOfs | LongInt | Ptr 32 bit a inizio prx IFD |

| Image File Directory (IFD) | | |
|----------------------------|--------------------------------|-----------------------------|
| Campo | Dimensione | Descrizione o valore |
| IFDCounter | Word | # IFDtags in questo IFD |
| IFDTagArray | Array[0..IFDCnt - 1] di IFDTag | Array di IFDtags |
| TIFFIFDOfs | LongInt | Ptr 32 bit a inizio prx IFD |

Il formato dei tag

| Campo | Dimensione | Descrizione o valore |
|--------------|-------------------|--------------------------------|
| Tag | Word | Valore del IDtag |
| DataType | Word | Byte, Short, Long, ASCII, etc. |
| DataCount | LongInt | # dati * Length(DataType) |
| DataPointer | LongInt | Ptr al blocco di dati |

Elaborazione Segnali e Immagini

19

Il formato GIF (Graphics Interchange Format)

- Sviluppato da CompuServe come formato per la rapida trasmissione di immagini via modem
- È copyrighted
- Primo standard risale al 87 (*87a*), poi 89 (*89a*)
- Supporta scala di grigio e colore a 1-8 bits (256 colori)
- Compressione basata su dizionario (LZW)
- Dati compressi organizzati in blocchi da 256 bytes

Elaborazione Segnali e Immagini

20

L'algorithmo di compressione LZW (Lempel, Ziv, and Welch)

- ❑ L'alfabeto non è noto a priori: la codifica avviene "man mano"...
- ❑ Percorrendo l'input il dizionario viene arricchito di volta in volta di nuove stringhe di bit a ciascuna delle quali si associa un codice
- ❑ Occorrenze future di tali stringhe vengono rimpiazzate dal codice (che è + corto della stringa stessa).
- ❑ Problemi legali tra Sperry e Unisys (-> PNG)

Elaborazione Segnali e Immagini

21

Meccanismo

- ❑ Si individuano le più brevi sottostringhe diverse: 10110101000 -> 1,0,11,01,010,00
- ❑ Ogni sottostringa è composta da un prefisso già codificato + un ulteriore bit (010 = 01+0), e viene codificata come: codice prefisso + extra bit
- ❑ Il codice è la posizione nel dizionario: per l' m -esima sottostringa il codice ha lunghezza $\log(m)$
- ❑ Per n sottostringhe diverse, la compressione è di $n \log(n)$

Elaborazione Segnali e Immagini

22

Esempio

Stringa in Input: 1 0 1 1 0 1 0 1 0 0 0 1 0



Dizionario

| Indice | Entry |
|--------|-------|
| 0 | ∅ |
| 1 | 1 |
| 2 | 0 |
| 3 | 11 |
| 4 | 01 |
| 5 | 010 |
| 6 | 00 |
| 7 | 10 |

| | | | | | | | |
|---|---|---|---|---|----|---|---|
| W | ∅ | ∅ | 1 | 0 | 01 | 0 | 1 |
| B | 1 | 0 | 1 | 1 | 0 | 0 | 0 |

Coppie:

(0,1) (0,0) (1,1) (2,1) (4,0) (2,0) (1,0)

Codifica:

0001 0000 0011 0101 1000 0100 0010

Elaborazione Segnali e Immagini

23

L'algoritmo

1. Inizializza il dizionario con la stringa vuota
2. **W** = sottostringa dell'input (più lunga possibile) contenuta nel dizionario
3. **B** = primo simbolo dell'input che segue **W**
4. Crea la coppia (codice **W,B**), dove "codice **W**" è la posizione di **W** nel dizionario
5. Aggiungi **W+B** al dizionario
6. Vai al punto 2

Elaborazione Segnali e Immagini

24

Confronto con altri metodi di compressione

- A differenza di Huffman, LZW sfrutta le dipendenze tra caratteri consecutivi
- LZW non necessita di due passaggi o "scan" dell'input. Ovvero, non necessita di conoscere l'alfabeto a priori: è un metodo *adattivo*.

Elaborazione Segnali e Immagini

25

Confronto con altri metodi di compressione

| | Huffman | Aritmetico | LZW |
|------------------------------------|-----------------------------|---------------------|----------------------------------|
| Probabilità | Note a priori | Note a priori | Non note |
| Alfabeto | Note a priori | Note a priori | Non note |
| Perdita dati | No | No | No |
| Dipendenza caratteri vicini | Non sfruttata | Non sfruttata | Sfruttata (miglior compressione) |
| Preprocessing | Costrz.albero $O(n \log n)$ | No | No |
| Codewords | 1 per simbolo | 1 per tutto l'input | 1 per sottostringa |
| Intuitività | abbastanza | No | Insomma... |

Confronto con altri metodi di compressione

| | LZW (unix <i>gzip</i>) | Huffman (unix <i>pack</i>) |
|------------------------------|----------------------------|--------------------------------|
| Html 25k (ascii, token) | 20% | 65% |
| Pdf 690k (binario) | 75% | 95% |
| ABCD 1.5k (ascii, random) | 33% | 28.2% |
| ABCD 500k (ascii, random) | 29% | 28.1% |

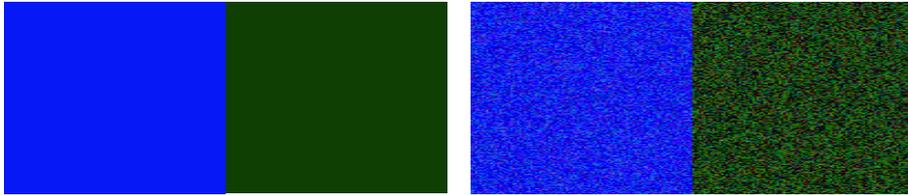
ABCD:
 $p_A=0.5$
 $p_B=0.25$
 $p_C=0.125$
 $p_D=0.125$

LZW è asintoticamente ottimale !!!

Il formato GIF: pros `n cons

- GIF: compressione lossless
- Distribuzione colore: GIF efficace per img con colore uniforme
- Profondità di colore: GIF 1-8 bit (2-256 colori), ottimizzazione della palette
- Interlacciato
- Trasparenza
- GIF animati

Colore uniforme e non...



Immagini di dimensione 200 x 200 pixel

| | | |
|---------|--------------------|---------|
| 2 bit | Profondità colore | 8 bit |
| 4 | Colori usati | 256 |
| 39,1 Kb | TIFF non compresso | 39,1 Kb |
| 10,1 Kb | GIF | 39,1 Kb |
| 1:15 | Compressione | 1:1 |

Elaborazione Segnali e Immagini

29

GIF o JPEG ?

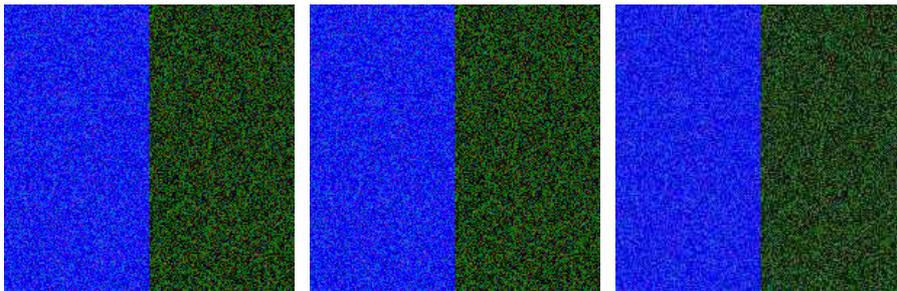


Immagine originale

GIF

JPG

| | | |
|----------------------|-------------------|----------|
| Qualità: | Ottima (lossless) | + o - OK |
| Compressione: | 1:1 | 1:17 |

Elaborazione Segnali e Immagini

30

In fotografia: GIF o JPG?

| | | |
|--|---|---|
|  | <p>GIF: 37,3 Kb 1:1 (dithering...)</p> |  <p>JPEG 1: 23 Kb 1:11</p> |
|  | <p>JPEG 2: 8,4 Kb 1:33</p> |  <p>JPEG 3: 2,9 Kb 1:96</p> |

Per le scritte: GIF o JPEG?

| | |
|---|--|
|  | <p>Scritta originale</p> |
|  | <p>JPG Il peggioramento della qualità è evidente. Lunghezza file: 2,3 Kb Compressione 1:29</p> |
|  | <p>GIF I contorni sono più nitidi. Il file è di poco più grande. Lunghezza file: 3,2 Kb Compressione: 1:5</p> |

Ottimizzazione della palette

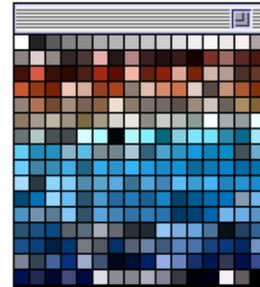
File Length
14,9 Kb

compressione:1:1



palette 256 colori
(8 bit)

adattata



File Length
9,1 Kb

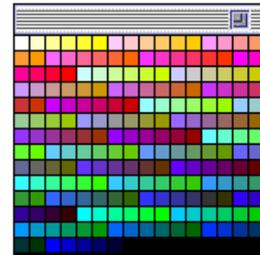
compressione:1:2

Questo è più
chiaro: meno
colori, meno
peso...



palette 216 colori
(8 bit)

web palette
(colori comuni
Mac-PC)



Ottimizzazione della palette

File Length
11 Kb

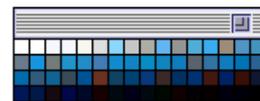
compressione:1:1

Meno colori, ma
nessuna
compressione.
Non conviene...



palette 64 colori
(6 bit)

adattata



File Length
6,8 Kb

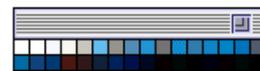
compressione:1:3

Usando meno
colori, anche la
compressione
aumenta.
La qualità è
ancora
accettabile.

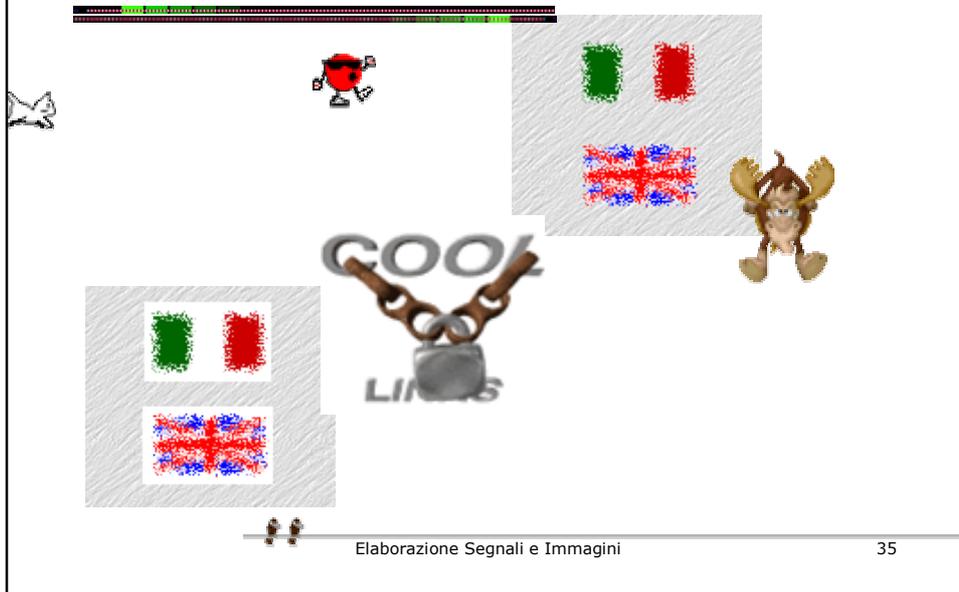


palette 32 colori
(5 bit)

adattata



Trasparenza e animazione



Elaborazione Segnali e Immagini

35