

SCHELETRIZZAZIONE

3.1 Introduzione

Uno degli obiettivi strategici della visione computerizzata è la ricerca e lo sviluppo di metodi flessibili per la descrizione delle forme. Un importante approccio per la rappresentazione di una forma strutturata di una regione è quello di ridurre quest'ultima ad un grafo.

Tale rappresentazione può essere ottenuta dalla valutazione del suo scheletro che contiene informazioni esplicitamente cruciali riguardo alla forma della regione di interesse.

La scheletrizzazione è, quindi, una trasformazione di un oggetto di interesse da una struttura complessa ad una struttura più semplice, che ne conservi e descriva tuttavia le caratteristiche principali.

Lo scheletro è, dunque, una rappresentazione costituita da un numero relativamente piccolo di elementi della forma dell'oggetto, in un certo senso quindi, lo scheletro è la struttura minima dell'oggetto stesso.

La posizione, l'orientamento e la lunghezza delle linee che compongono lo scheletro sono rappresentative di quelle di cui l'immagine è costituita e questa schematizzazione risulta di aiuto per semplificare gli algoritmi di analisi e il riconoscimento d'immagine.

D'altro canto è importante ricordare che, anche per l'uomo, la percezione dell'immagine di un carattere avviene come una serie di tratti rettilinei o curvi, il cui spessore è solitamente uniforme e trascurabile rispetto alla lunghezza.

Una definizione concisa di scheletro, ed universalmente accettata, è data in [1] dove lo scheletro è definito come il luogo dei centri delle massime circonferenze contenute nella forma. Questa rappresentazione dà origine ad un descrittore di forma molto interessante perché gode di una serie di proprietà importanti.

Innanzitutto lo scheletro è un buon descrittore di forma, nel senso che gode delle proprietà di invarianza per traslazione, rotazione, scalamento. Inoltre esso riflette importanti caratteristiche visive della forma che rappresenta, come la simmetria e la complessità. Lo scheletro, in aggiunta, è un buon descrittore per rappresentare delle regioni che possono essere deformate e può essere anche usato come base per una forma di "compressione" della informazione visiva.

Da quanto detto si evince la potenza di questo strumento di rappresentazione ma come valutarne la bontà?

Uno scheletro, e quindi un metodo di scheletrizzazione, è tanto migliore quanto più è in grado di descrivere una forma senza perdere informazioni importanti e, possibilmente, permettendo di ricostruire in modo efficiente e preciso la forma iniziale.

Un'altra proprietà auspicabile è che il metodo di scheletrizzazione sia il più possibile insensibile ai rumori, nel senso di non produrre delle appendici non desiderabili alla forma ideale dello scheletro.

In ultimo è importante considerare il peso dal punto di vista computazionale del metodo.

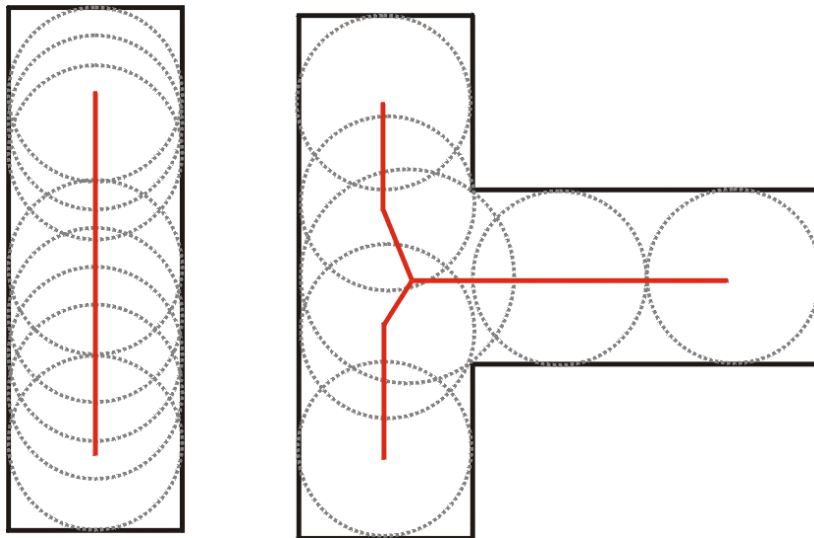


Figura 3.1 : Scheletro costruito a partire dalla definizione di luogo dei centri delle massime circonferenze contenute nella forma.

L'inconveniente di questa definizione è che lo scheletro presenta delle distorsioni in coincidenza dell'incrocio dei tratti a tal proposito si veda la Figura 3.1. Tali distorsioni non sono solo tipiche della definizione di scheletro ma sono visibili in molte tecniche di costruzione dello stesso. Questa particolarità rende, generalmente, necessaria una fase di correzione delle distorsioni.

3.2 Una panoramica sui metodi classici

Gli approcci per ottenere lo scheletro di un oggetto contenuto in un'immagine sono molteplici. E' questo infatti un argomento molto considerato dai ricercatori e ogni anno

vengono sempre proposti nuovi algoritmi o nuovi approcci al problema. Dovendo indicare per sommi capi i principali metodi ne presentiamo solo alcuni.

Tutti, fatta eccezione per il metodo conosciuto come *ricerca dell'asse mediano*, partono da un'immagine precedentemente binarizzata per poi individuarne le componenti principali ovvero togliere gli "strati" esterni dell'oggetto. Per questo motivo l'approccio è considerato di tipo morfologico – erosivo.

Si metterà in evidenza come questi metodi, sebbene estremamente semplici a livello computazionale ed implementativo, non offrano sempre dei buoni risultati. Questo perché l'immagine di partenza è binarizzata, tale processo permette di separare il background dall'oggetto tuttavia riduce le informazioni concernenti l'oggetto stesso.

Uno dei metodi più comuni di determinazione dello scheletro è stato individuato da Blum (1967) nella trasformazione dell'asse mediano della regione MAT, Median Axes Transformation. Formalmente, per ogni punto p appartenente ad una regione R di bordo B , si determina il punto in B ad esso più vicino. Se esiste più di un punto in B avente distanza minima da p allora si dice che p appartiene all'asse mediano o scheletro di R .

Il criterio di *minima distanza* o *vicinanza* adottato per la corrispondenza fra $p \in R$ e $b \in B$ influenza la determinazione dello scheletro.

Solitamente si utilizza la formula della distanza euclidea con risultati soddisfacenti.

Come tutti i metodi che comportano il calcolo di distanze anche l'esecuzione diretta della MAT è proibitiva da un punto di vista computazionale. Essa implica il calcolo per ogni punto $p \in R$, della distanza di tutti i punti $b \in B$. Le alternative proposte migliorano normalmente l'efficienza computazionale: esse sono basate su cicli iterativi di cancellazione del contorno dell'oggetto rispettando alcuni vincoli prestabiliti, quali la non rimozione delle estremità, la non distruzione della connettività dell'oggetto, la non produzione di eccessiva erosione dell'oggetto.

Presentiamo l'algoritmo sviluppato da Zhang e Suen (1984) per assottigliare una regione appartenente ad un'immagine precedentemente binarizzata.

Nel corso della trattazione si assumerà che i punti della regione di interesse abbiano valore 1 mentre quelli del background 0. Il metodo consiste nella ripetizione di un ciclo costituito da due passi applicato ai punti di contorno di una data regione, dove un punto di contorno è un pixel con valore 1 e avente un punto vicino nell'otto-intorno uguale a 0.

P9	P2	P3
P8	P1	P4
P7	P6	P5

Figura 3.2

Nominando i punti intorno al generico pixel P_1 seguendo l'esempio in Figura 3.2 si può procedere nel seguente modo:

Passo 1:

I pixel intorno al punto P_1 vengono marcati al fine di essere cancellati in seguito se le seguenti condizioni sono soddisfatte:

- a) $2 \leq N(p_1) \leq 6$
- b) $S(p_1) = 1$
- c) $p_2 \cdot p_4 \cdot p_6 = 0$
- d) $p_4 \cdot p_6 \cdot p_8 = 0$

Dove $N(p_1)$ è il numero di pixel nell'intorno di p_1 non uguali a zero, cioè

$$N(p_1) = p_2 + p_3 + p_4 + \dots + p_9$$

e $S(p_1)$ è il numero di 0-1 transizioni nella sequenza ordinata di $p_2, p_3 \dots p_8, p_9$.

Per esempio $N(p_1) = 4$ e $S(p_1) = 3$ come riportato in Figura 3.3

0	0	1
1	P ₁	0
1	0	1

Figura 3.3

Passo 2:

Le condizioni (a) e (b) rimangono invariate, ma la condizione (c) e (d) diventano:

- c') $p_2 \cdot p_4 \cdot p_8 = 0$
- d') $p_2 \cdot p_6 \cdot p_8 = 0$

Il Passo 1 è applicato a tutti i pixel appartenenti al bordo della regione binaria in considerazione. Se una o più condizioni da (a) a (d) risultano violate il valore del punto in questione non viene cambiato. Se invece tutte le condizioni sono soddisfatte il punto è segnalato come candidato all'eliminazione. È importante notare tuttavia che il punto non viene cancellato finché tutti i punti del bordo non sono stati esaminati. Questo previene i cambiamenti nella struttura durante l'esecuzione dell'algoritmo. Dopo aver applicato il Passo 1 a tutti i punti del bordo tutti quelli candidati alla cancellazione vengono eliminati. Quindi viene applicato il Passo 2 nella stessa maniera in cui è stato applicato il Passo 1. Un'iterazione dell'algoritmo è costituita da questa sequenza di operazioni:

- 1) applicazione del Passo 1 e individuazione dei punti da cancellare;
- 2) cancellazione dei punti precedentemente marcati;
- 3) applicazione del Passo 2 e individuazione dei punti da eliminare;
- 4) cancellazione dei punti marcati.

Questa procedura di base è applicata iterativamente finché tutti i punti eccedenti non vengono cancellati lasciando come risultato lo scheletro della regione. L'algoritmo termina nel momento in cui non ci sono più punti da cancellare.

Si illustra ora il significato delle condizioni (a), (b), (c), (d).

La condizione (a) è violata quando il contorno del punto p_1 conta un solo punto nell'otto-intorno uguale a 1. In tal caso p_1 sarebbe il punto terminale di un segmento e quindi non deve essere cancellato. Nello stesso modo se p_1 ha più di sette elementi nell'otto-intorno uguali ad 1 non può essere cancellato per non causare un'eccessiva erosione della regione.

La condizione (b) è violata quando p_1 è un punto di una linea costituente lo scheletro. Questa condizione previene la disconnessione dei segmenti dello scheletro durante l'operazione di assottigliamento. Le condizioni (c) e (d) sono soddisfatte entrambe solo se p_4 o p_6 è uguale a 0 oppure se p_3 e p_8 sono entrambi 0.

Questo significa che se (a), (b), (c) e (d) sono soddisfatte p_1 appartiene al bordo sinistro dell'oggetto oppure al bordo inferiore oppure è uno spigolo superiore sinistro. Si veda a tal proposito la Figura 3.4.

0	0	0
0	1	1
0	1	1

Sono soddisfatte le condizioni (a) $N(p_1) = 3$ (b) $S(p_1) = 1$.

Vale inoltre $p_2 = p_8 = 0$. Il punto centrale è uno spigolo superiore sinistro e va eliminato

1	1	0
1	1	0
1	1	0

Sono soddisfatte le condizioni (a) $N(p_1) = 3$ (b) $S(p_1) = 1$.

Vale inoltre $p_4 = 0$. Il punto centrale è parte di un bordo sinistro e va eliminato

1	1	1
1	1	1
0	0	0

Sono soddisfatte le condizioni (a) $N(p_1) = 3$ (b) $S(p_1) = 1$.

Vale inoltre $p_6 = 0$. Il punto centrale è parte di un bordo inferiore e va eliminato

Figura 3.4

In modo del tutto analogo le condizioni (a), (b), (c') e (d') se tutte soddisfatte permettono di eliminare i punti appartenenti al bordo inferiore, al bordo destro o agli spigoli inferiori sinistri di un oggetto.

Si presentano ora alcuni risultati prodotti dall'algoritmo di scheletrizzazione sulle immagini precedentemente binarizzate.

Nelle Figure 3.2 e 3.3 sono presentate due scheletrizzazioni applicate all'immagine binarizzata prima col metodo Iterativo e poi con il metodo di Kapur. Si noti quanto accennato in precedenza in merito alla fase di binarizzazione. Se questa, infatti, non è sufficientemente accurata i risultati della scheletrizzazione non sono soddisfacenti in quanto non rappresentano efficientemente la forma minima della regione di interesse, in questo caso i caratteri componenti la scritta.

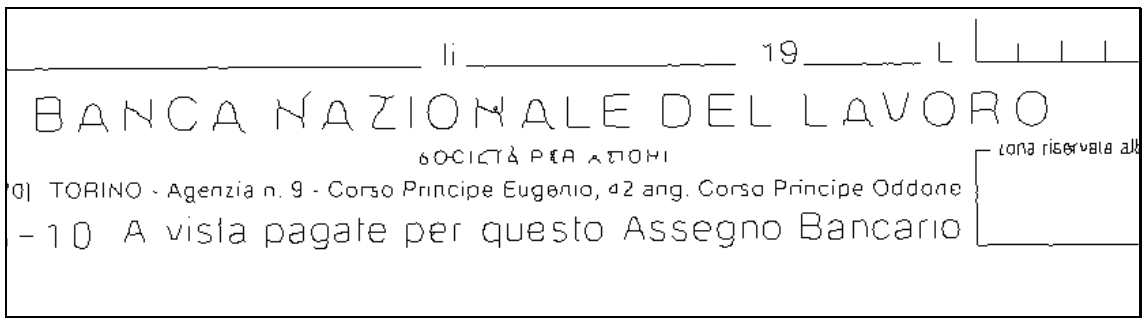


Figura 3.5 : Scheletrizzazione con algoritmo di Zhang su immagine sogliata con metodo iterativo
(soglia 173)

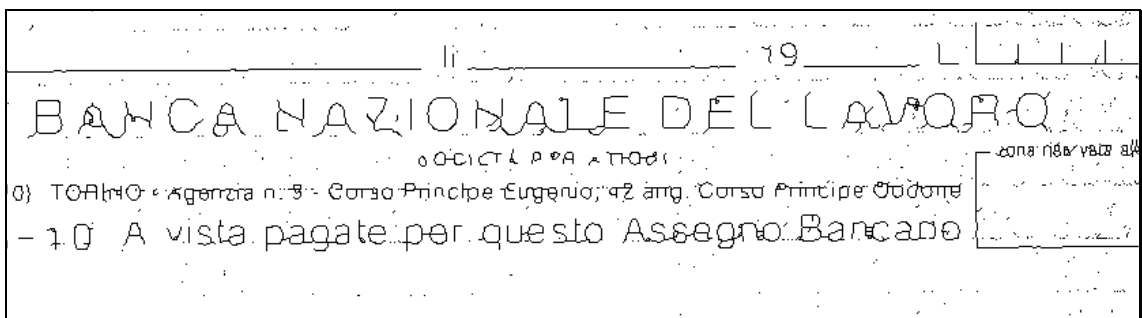


Figura 3.6 Scheletrizzazione con algoritmo di Zhang su immagine sogliata con metodo di Kapur
(soglia 187)

Per meglio comprendere il comportamento dell'algoritmo di scheletrizzazione se ne osservi l'effetto su una porzione della immagine precedente. Nella sequenza di figure si potrà notare il comportamento erosivo del metodo.



Fig3.7: Immagine originale



Figura 3.8: Immagine sogliata (soglia 173)



Figura 3.9: Iterazione 1



Figura 3.10: Iterazione 2



Figura 3.11: Iterazione 3



Figura 3.12: Iterazione 4



Figura 3.13 : Iterazione 5



Figura 3.14: Iterazione 6

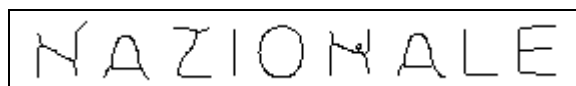


Figura 3.15 :Iterazione 7



Figura 3.16 :Risultato finale in rapporto alla immagine di partenza.

Per giungere alla immagine finale sono bastate 7 iterazioni dell'algoritmo le quali sono state portate a termine in 18146 μ s. Si può concludere che tale algoritmo sia particolarmente efficiente in termini computazionali, anche se il risultato finale è soggetto ad alcune limitazioni. In primo luogo si può osservare che la scheletrizzazione è veramente soddisfacente solo in alcuni casi. Se infatti i caratteri sono abbastanza

regolari e squadrati i risultati sono particolarmente buoni come nel caso delle lettere L, I, O, E.

Se invece si osservano le lettere N, Z, A, si nota subito che i raccordi tra i tratti obliqui non sono rappresentati in modo corretto, in quanto si viene a perdere la informazione di continuità del tratto in corrispondenza degli incroci. Si veda la Figura 3.16 dove vengono raffrontati il risultato ottenuto e quello ideale.

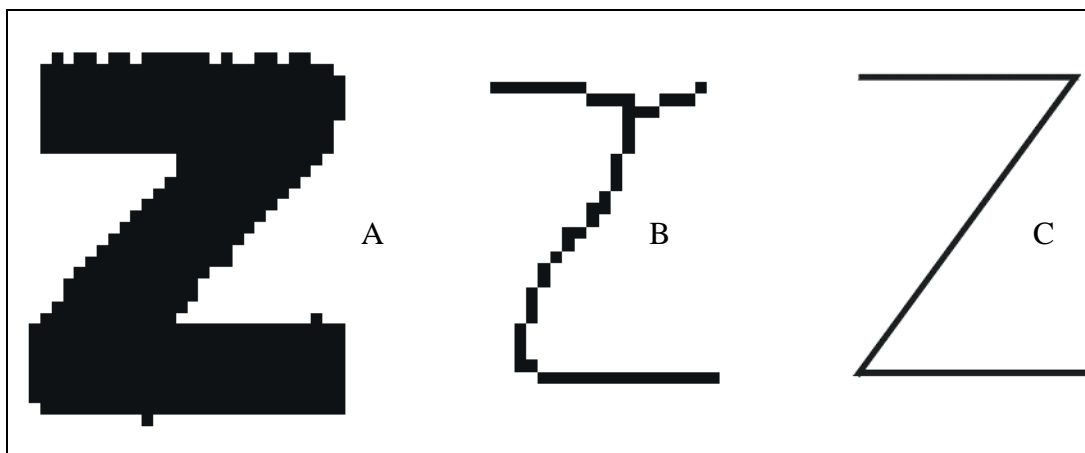


Figura 3.17 : (A)Ingrandimento della immagine sogliata, (B) scheletrizzazione, (C) scheletro ideale della lettera

Si noti come, nelle varie fasi della scheletrizzazione, le componenti diagonali del carattere vengano distorti. Questo fenomeno è causato dalla mancata considerazione delle componenti diagonali nelle maschere di erosione. Tale azione viene infatti eseguita solo sulle componenti fondamentali verticali ed orizzontali.

Osservando, inoltre, il carattere N nella Figura 3.15 si nota come basti una piccola alterazione sul contorno della lettera in questione per causare una errata scheletrizzazione.

Esistono in letteratura un gran numero di varianti a quest' algoritmo di scheletrizzazione le quali producono risultati migliori o ottimizzano i tempi di esecuzione. Tali varianti, tuttavia, soffrono della limitazione intrinseca dell'approccio erosivo, ovvero producono distorsioni più o meno evidenti rispetto alle linee guida che identificano lo scheletro ideale del carattere o dell'oggetto in esame.

Parallelamente al miglioramento di queste tecniche se ne sono sviluppate altre che partono da un diverso approccio al problema.

Solo al fine di presentare un altro tipo di proposta di costruzione dello scheletro si consideri la tecnica della *veinerization* [2] (tecnica delle venature), esso viene costruito per successive eliminazioni non dei contorni ma dei possibili cammini strutturati che partono dai bordi della immagine e convergono verso l'asse principale dell'oggetto.

In seguito si presenterà in modo più approfondito una tecnica sicuramente più fine di quella ottenuta per erosione basata su un approccio matematico – topografico della immagine di partenza considerata non più in forma binarizzata ma come immagine a toni di grigio. Questo procedimento risulterà meno sensibile sia agli errori tipici dell'algoritmo di scheletrizzazione di Zhang, sia ai disturbi generici di natura gaussiana.

Si include una possibile implementazione dell'algoritmo di Zhang usato nella valutazione del metodo nel contesto di questo lavoro.

```
//-----
// Function name      : CBinarize::ZhangSk   (01/11/2000)
// Description       : Metodo di scheletrizzazione di Zhang
// Return type       : BOOL
// Argument          : BYTE nBGColor, int * nIterations (colore
//                   background, numero di iterazioni)
// Note              : m_plmg è un puntatore ai BYTE della immagine
//                   binarizzata di cui si vuole costruire lo scheletro
//                   m_nX , m_nY, m_nW sono le dimensioni della immagine
//-----

// Movimento della maschera di analisi
#define DXP p1++, p2++, p3++, p4++, p5++, p6++, p7++, p8++, p9++
#define SOTTOP p1 += m, p2 += m, p3 += m, p4 += m, p5 += m, p6 += m, p7 += m, p8
+= m, p9 += m
#define DXQ q1++, q2++, q3++, q4++, q5++, q6++, q7++, q8++, q9++
#define SOTTOQ q1 += m, q2 += m, q3 += m, q4 += m, q5 += m, q6 += m, q7 += m,
q8 += m, q9 += m

BOOL CBinarize::ZhangSk(BYTE nBGColor, int *nIterations)
{
    // test generici
    _ASSERT(m_plmg != NULL);
```

```
if (m_plmg == NULL) return FALSE ;
_ASSERTE( (nBGColor == 0) || (nBGColor == 255)) ;
if ((nBGColor != 0) && (nBGColor != 255)) return FALSE ;

// Test immagine binarizzata + Creazione immagine Copia con 0 e 1
LPBYTE pCopy = (LPBYTE) new BYTE[m_nW * m_nY] ;
_ASSERTE(pCopy != NULL) ;
if (pCopy == NULL) return FALSE ;
ZeroMemory(pCopy, m_nW * m_nY) ;
LPBYTE q = m_plmg ;
LPBYTE p = pCopy ;
int k = m_nW - m_nX ;
int y, x ;
for (y = 0; y < m_nY; y++, q += k, p += k)
  for (x = 0; x < m_nX; x++, q++, p++)
  {
    if ((*q != 255) && (*q != 0))
    {
      delete pCopy ;
      return FALSE ;
    }
    else
      *p = (*q ^ nBGColor) >> 7;
  }

BYTE nMark = 'M';
// Puntatori agli elementi della matrice
LPBYTE p1, p2, p3, p4, p5, p6, p7, p8, p9 ;
LPBYTE q1, q2, q3, q4, q5, q6, q7, q8, q9 ;
int m = k + 2;

int nDeleted = 0 ;
int nTime = 0 ;
BOOL fEnd = FALSE ;
for(;;)
{
  // configurazione sulla immagine copia
  p1 = pCopy + m_nW + 1 ;
  p2 = p1 - m_nW ; p3 = p2 + 1 ; p4 = p1 + 1 ;
  p5 = p4 + m_nW ; p6 = p5 - 1 ; p7 = p6 - 1 ;
  p8 = p1 - 1 ; p9 = p2 - 1 ;

  // configurazione sulla immagine originale
```

```

q1 = m_plmg + m_nW + 1 ;
q2 = q1 - m_nW ; q3 = q2 + 1 ; q4 = q1 + 1 ;
q5 = q4 + m_nW ; q6 = q5 - 1 ; q7 = q6 - 1 ;
q8 = q1 - 1 ; q9 = q2 - 1 ;

for (y = 1; y < m_nY - 1; y++, SOTTO, SOTTOQ)
  for (x = 1; x < m_nX - 1; x++, DX, DXQ)
  {
    if (*p1 == 0)
      continue ;
    else
    {
      int N = *p2 + *p3 + *p4 + *p5 + *p6 + *p7 + *p8 + *p9 ;
      if ((N < 2) || (N > 6))
        continue ;
      int S = abs(*p3 - *p2) + abs(*p4 - *p3) + abs(*p5 - *p4) + abs(*p6 - *p5) + abs(*p7 -
*p6)
      + abs(*p8 - *p7) + abs(*p9 - *p8) + abs(*p2 - *p9) ;
      if (S != 2)
        continue ;
      if (!(nTime & 1)) // Step 1
      {
        if ((*p2 & *p4 & *p6) != 0)
          continue ;
        if ((*p4 & *p6 & *p8) != 0)
          continue ;
      }
      else // Step 2
      {
        if ((*p2 & *p4 & *p8) != 0)
          continue ;
        if ((*p2 & *p6 & *p8) != 0)
          continue ;
      }
      // mark
      *q1 = nMark ;
    }
  }
  // Deleting
  nDeleted = 0 ;
  p = pCopy ;
  q = m_plmg ;
  for (y = 0; y < m_nY; y++, q += k, p += k )

```

```
for (x = 0; x < m_nX; x++, q++, p++)
{
    if (*q == nMark)
    {
        nDeleted ++ ;
        *q = nBGColor ;
        *p = 0 ;
    }
}
// Test uscita
if ((nDeleted == 0) && fEnd) break ;
else if ((nDeleted == 0) && !fEnd) fEnd = TRUE ;
else fEnd = FALSE ;
nTime ++ ;
}
// Release resource
delete pCopy ;
if (nIterations != NULL)
    *nIterations = (nTime + 1) / 2 ;
return TRUE ;
}
```

Riferimenti bibliografici

[1] H.Blum. “*A transformation for extracting new parameter of shape*”, MODELS FOR THE PERCEPTION OF SPEECH AND VISUAL FORM. MIT Press, 1967.

[2] Marc Pierrot Deseilligny, Georges Stamon, Ching Y. Suen, “*Veinerization: A new shape description for Flexible Skeletonization* ”, IEEE TRANSACTION ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL 20, NO 5, MAY 1998

[3] King-Sun Fu, Rafael C. Gonzalez, C.S. Gorge Lee, “*ROBOTICA*” ,McGraw-Hill Libri Italia s.r.l. 1989

[4] Milang Sonka, Vaclav Hlavac, Riger Boyle “*Image Processing, Analysis, and Machine Vision - Second Edition*” PWS PUBLISHING