

Consiglio Nazionale delle Ricerche

**Un algoritmo per la codifica e  
la descrizione automatica dei  
contorni in immagini digitali**

130

***CNUCE***

Divisione Servizio Elaborazione Dati

A cura di : G. Fresta

CNUCE - Istituto del CNR - Pisa

F. Massimo - P. Mussio - S. Zambon - M. Zuccon

LFCTR - Milano via Celoria, 16

Copyright Luglio 1977

by - CNUCE - Pisa

Istituto del Consiglio Nazionale delle Ricerche

UN ALGORITMO PER LA CODIFICA E  
LA DESCRIZIONE AUTOMATICA DEI  
CONTORNI IN IMMAGINI DIGITALI

+ G.FRESTA

++ F.MASSEMO, P.MUSSIO, S.ZAMBON, M.ZUCCON

Gli autori (+) ringraziano la Sig.ra E. Lofrese per la  
fattiva collaborazione che ha consentito la redazione del  
presente lavoro.

+ CNUCE Istituto del CNR PISA  
++ LFCTR MILANO Via Celoria 16

## S O M M A R I O

In questo rapporto e' presentato un algoritmo per la descrizione dei contorni di immagini bicolori basato su un L-sistema bidimensionale. L'algoritmo ricava la totalita' dell'informazione relativa ai punti di contorno dell'immagine, assegnando ad ogni punto una stringa di codifica che permette di risalire alla situazione di ciascun punto di contorno in relazione alle sue vicinanze. Vengono presentate applicazioni su figure regolari e figure rumorose. La realizzazione in linguaggio APL e' mostrata e commentata.

## I N D I C E

1- Introduzione	p. 1
2- Descrizione formale del metodo	p. 2
3- La stringa di codifica	p. 4
4- La descrizione dei contorni	p. 7
5- Implementazione in linguaggio APL	p. 9
Esempio 1	p. 13
Esempio 2	p. 15
Esempio 3	p. 20
Esempio 4	p. 26
Bibliografia	p. 29
Lista delle funzioni APL	p. 30

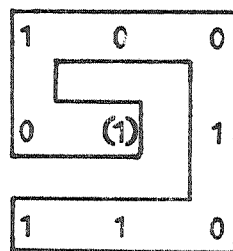
## 1- Introduzione

In questo rapporto e' presentato un algoritmo per la descrizione dei contorni di immagini bicolori basato su un L-sistema bidimensionale. La sua realizzazione in linguaggio APL e' mostrata e commentata.

Relativamente ai fini del nostro lavoro sara' detta immagine bicolore una matrice bidimensionale i cui elementi sono i simboli 0 e 1. Tale matrice puo' essere generata casualmente o puo' costituire il risultato di una serie di trattamenti cui un'immagine reale e' stata sottoposta per renderne possibile l'analisi ed il riconoscimento automatici (1).

L'algoritmo si propone di analizzare il variare dei simboli (0 ed 1), evidenziando quelli che presentano uno o piu' mutamenti nelle loro vicinanze: questi sono detti punti di contorno. Piu' precisamente, se per ogni punto - elemento della matrice binaria - si considera la stringa bidimensionale formata dal simbolo che rappresenta il punto e da quelli immediatamente vicini (cfr. Fig. 1), ciascun punto e' considerato di contorno se la stringa bidimensionale ad esso associata presenta almeno una variazione (01; 10).

Fig. 1



stringa

bidimensionale

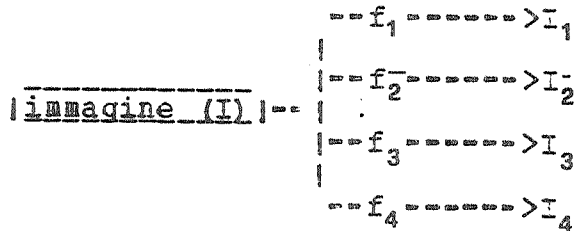
E' stato scelto il criterio di mantenere inalterata la totalita' dell'informazione associata ad ogni punto, in modo da procedere solo in seguito alle semplificazioni richieste dai particolari metodi di riconoscimenti adottati (2). Tale scopo e' stato raggiunto individuando quali punti sono di contorno e come lo sono, cioe' quali e quante variazioni presenta la stringa associata ad ogni punto di contorno.

Per i criteri sui quali si basa, tale algoritmo puo' facilmente venire esteso alla descrizione dei contorni di immagini n-arie (a piu' colori) e/o n-dimensionali.

2- Descrizione formale del metodo

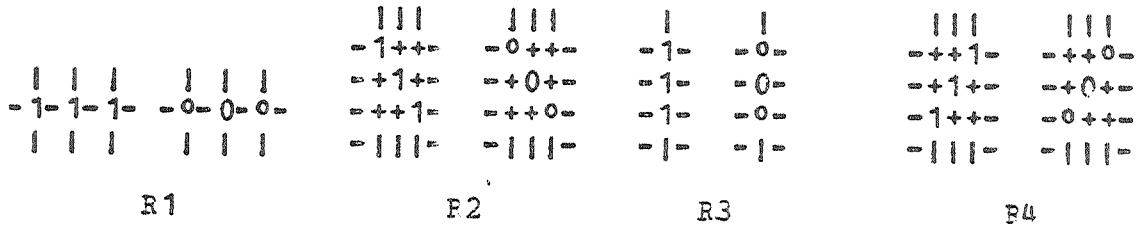
L'algoritmo per l'analisi dell'immagine e' basato su quattro funzioni espresse nella forma di L-sistemi (3). L'alfabeto dei simboli e' dato dall'insieme 0,1 ; la stringa assioma di partenza e' la matrice bidimensionale. Le quattro funzioni sono descritte da quattro regole grafiche di riscrittura e una regola di applicazione. Applicando le funzioni sull'immagine si ottengono quattro nuove matrici la cui unione da' l'immagine con i contorni.

Fig. 2



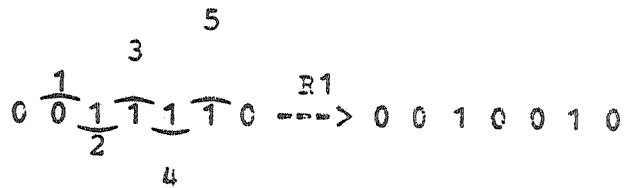
Preso una griglia di dimensioni opportune che su ogni nodo ha uno dei simboli, la regola di riscrittura la trasforma in una nuova griglia che nei nodi avra' ancora uno dei simboli (il segno '0' e' sostitutivo di 0 o di 1) (4). La regola di applicazione stabilisce che la griglia viene sovrapposta all'immagine in modo che vi sia sempre interamente contenuta. Se ogni simbolo che sta sul nodo della griglia e' lo stesso dell'immagine, allora si applica la regola di riscrittura corrispondente, altrimenti l'immagine rimane invariata. Ogni regola di riscrittura viene applicata in tutte le posizioni possibili dell'immagine nello stesso istante. In questo senso chiamiamo il sistema formale cosi' descritto un L-sistema. Le quattro funzioni sono descritte dalle seguenti regole che stabiliscono quando il punto contrassegnato dal simbolo 1 e' di contorno.

Fig. 3



La' dove si verifica una delle situazioni presentate nella prima griglia di ogni regola, questa stabilisce che la posizione centrale assume il simbolo 0, mentre le altre assumono 0 o 1 in dipendenza dal contesto. L'applicazione della regola di riscrittura non avviene in sequenza, ma contemporaneamente su tutta l'immagine (cfr. Esempio 2).

Fig. 4



In Fig. 4 la R1 e' applicabile solo nelle posizioni 3 e 4.

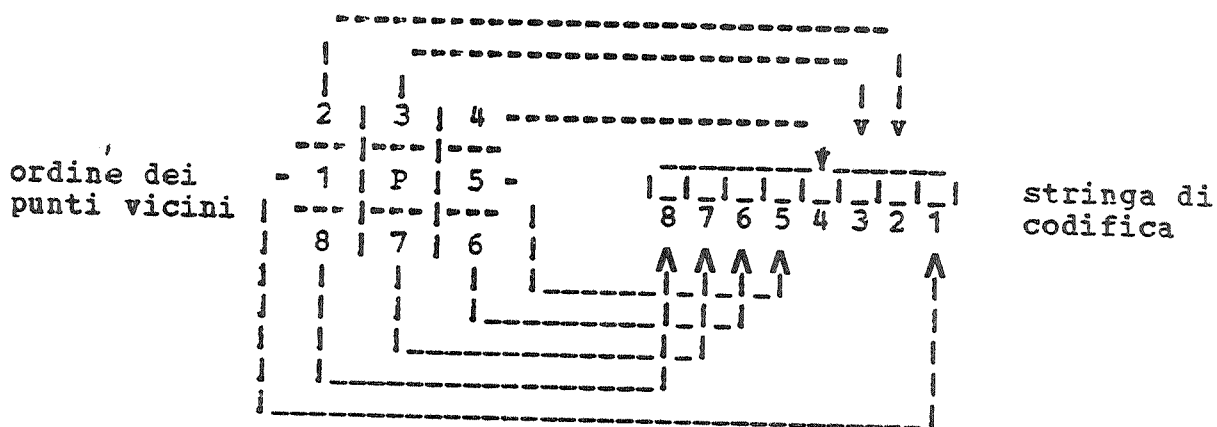


### 3- La stringa di codifica

Dalle matrici prodotte dalle funzioni dell'L-sistema si possono ricavare anche le caratteristiche di ciascun punto di contorno - come un punto e' di contorno. In tali matrici stanno infatti i simboli che ciascun punto dell'immagine assume dopo l'applicazione di una delle regole. Ricavando ordinatamente tale informazione e ordinandola in una stringa detta di codifica, si avra' la descrizione della situazione di ciascun punto rispetto ai vicini cosi' come e' stabilita dalle quattro regole.

Per ottenere un'informazione completa e' pero' necessario conoscere per ogni vicino cio' che la regola stabilisce. Questo risultato e' raggiunto spezzando in due momenti l'applicazione di ciascuna regola, considerando prima il vicino a destra (o in alto), poi quello a sinistra (o in basso). La stringa di codifica cosi' ottenuta e' composta di otto elementi ognuno dei quali mostra quale simbolo assume il punto considerato (P) rispetto al vicino che occupa la posizione corrispondente secondo l'ordine illustrato in Fig. 5.

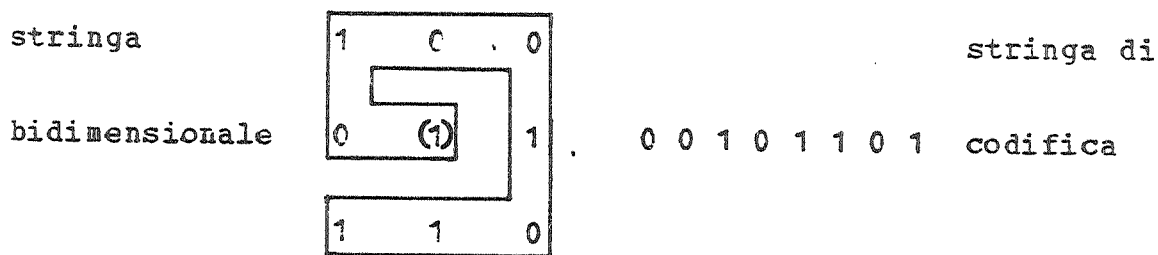
Fig. 5



Si puo' notare come la stringa di codifica derivi dalla stringa bidimensionale. Gli otto elementi della stringa di codifica sono infatti i simboli prodotti dall'applicazione delle regole sul punto che nell'analisi viene considerato nel contesto dei vicini stabilito dalla stringa bidimensionale.

La figura seguente mostra la stringa di codifica relativa al punto la cui stringa bidimensionale sta in Fig. 1. Poiche' si cerca la variazione rispetto al simbolo 1 e tale simbolo (0) compare piu' volte nella stringa di codifica, il punto analizzato e' un punto di contorno.

Fig. 6



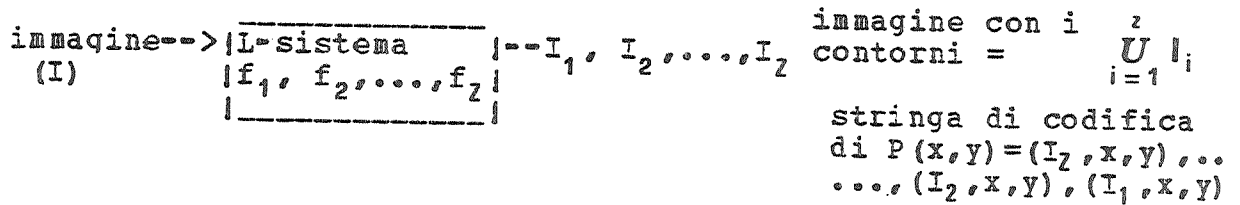
E' immediato osservare come nel nostro caso si verifichi una coincidenza tra simboli prodotti dall'applicazione delle regole e valori risultanti dal confronto logico tra il punto considerato e i propri vicini. La stringa di simboli e' una stringa di valori effettivi nell'ambito della logica binaria.

Tale coincidenza consente ovviamente una notevole semplificazione all'atto dell'implementazione dell'algoritmo, ma non e' necessaria. Nel caso di immagini con n-simboli, potendo essere i simboli segni arbitrari, si puo' anche non conservare l'omogeneita' tra simboli e valori, continuando ad usare procedure di calcolo basate sulla consueta logica binaria. Tuttavia mantenere la corrispondenza tra L-sistema e tipo di logica usata nell'applicarlo e nell'implementarlo, puo' essere utile se ci si vorra' servire delle proprieta' di questa a vantaggio di quello, come e' mostrato dalla semplice applicazione presentata in questo rapporto.

In generale dunque, data un'immagine di dimensioni qualsivoglia con un arbitrario numero di simboli, l'L-sistema la analizza mediante tante funzioni quanti sono i vicini considerati nella stringa bidimensionale. Ogni funzione produce una matrice delle stesse dimensioni della immagine; tale matrice costituisce il risultato dell'applicazione della regola di riscrittura relativa alla funzione. Da queste matrici e' possibile sia ricostruire l'immagine con i contorni evidenziati, sia ottenere la stringa di codifica di ogni punto (P) di contorno.

In Fig. 7 (x,y) sono le coordinate dell'ipotetico punto di contorno; (z) e' il numero complessivo dei vicini stabiliti dalla stringa bidimensionale adottata.

Fig. 7



L'ordine inverso della codifica del punto di contorno rispetto all'ordine delle matrici date in uscita e' stato adottato per sfruttare un'altra utile coincidenza della stringa di codifica. Essa infatti puo' essere interpretata come un numero codificato in base binaria che rappresenta univocamente su otto posizioni tutti i possibili tipi di punto - nel nostro caso  $2^8=256$ . L'ordine adottato consente di ricavare con immediatezza alcune proprieta' del punto in esame. Per esempio, tutte le potenze di 2 indicano che il punto e' di contorno per un solo vicino, e la potenza indica la posizione del vicino. L'esempio 3 illustra come molte altre informazioni simili a questa possono essere ricavate dalla stringa di codifica.

4- La descrizione dei contorni

Per la completa relazione che la stringa di codifica stabilisce tra tutti i punti dell'immagine, e' possibile ottenere, oltre che la messa in rilievo, anche una descrizione dei contorni punto per punto. Potenzialmente, adottando nel nostro caso 256 simboli, ciascun punto dell'immagine potrebbe essere caratterizzato, rispetto alla situazione nelle sue vicinanze, secondo una delle 256 stringhe di codifica possibili. Tuttavia risulta in genere piu' utile riunire tra loro quelle combinazioni cui si vuole in particolare prestare attenzione.

Si e' gia' detto come tutte le potenze di 2, ottenute dalla interpretazione della stringa come numero binario, individuano un preciso tipo di punto di contorno: vertici di angoli retti del tipo:

+++  
-⊕+  
+++

comunque orientati, eccetto il caso in cui il punto si trovi tra due punti a loro volta vertici di angolo retto; per es:

++-++  
++⊕++  
+++++

(+ e' il simbolo rispetto a cui si cerca la variazione; ⊕ e' il punto considerato).

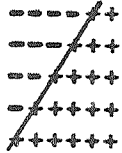
,Anche situazioni del tipo:

-++  
-⊕+  
-++

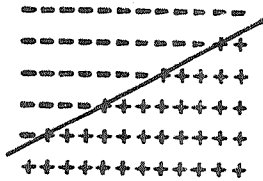
comunque orientate, che caratterizzano punti di retta, sono subito individuate trovando le stringhe di codifica interpretabili - in decimale - secondo la progressione geometrica 7 14 28 56 112 224 e poi all'inverso - calcolando il complemento da 224 a 255 - 224 193 131 7.

Se poi si passa ad analizzare relazioni tra stringhe, si puo' facilmente stabilire per es: se un'immagine contiene figure chiuse, usando come criterio di chiusura la verifica dell'esistenza, per ogni punto di contorno, del punto con eguale indice di riga, colonna o diagonale, che sia di contorno per il vicino simmetrico (vicini simmetrici sono quelli nelle posizioni 1-5, 2-6, ecc. (cfr. Fig. 5)). Tale condizione e' soddisfatta dalla presenza del simbolo convenuto nella posizione della stringa che riporta il risultato del confronto con il vicino simmetrico.

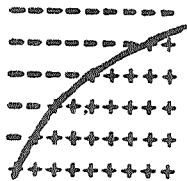
Oppure situazioni del tipo:



si puo' stabilire di considerarle rette oblique dopo aver verificato la regolarita' del susseguirsi di punti di retta e punti di vertice di angolo retto rivolti all'"interno". Retta obliqua con diversa inclinazione puo' essere considerata una situazione del tipo



Regolarita' del tipo:



si possono individuare a partire dalle stringhe di codifica che, indicando come ogni punto di contorno sta sul contorno - quali sono le posizioni dei vicini con simbolo diverso (cfr. Fig. 5)-, permettono la formazione di sequenze del tipo 123;2;1234;23;2;1234;234;23;2;1234; in grado appunto di descrivere la regolarita' del contorno presentato in figura.

Una opportuna considerazione della stringa di codifica, singola o posta in relazione con le altre, consente quindi di ricavare interessanti informazioni circa il problema dei contorni in immagini digitali, e di costruire algoritmi che, basandosi sull'insieme delle stringhe, sono in grado di fornire tali informazioni in modo automatico (5).

Altre possibili descrizioni dei contorni ricavabili dalle stringhe di codifica sono mostrate nell'esempio 3.

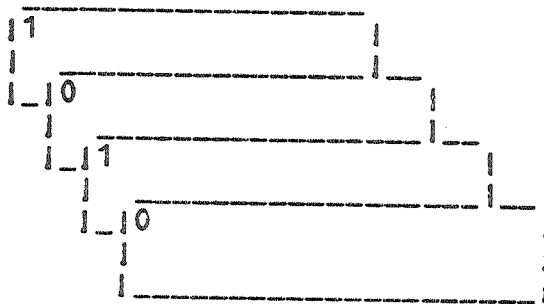
### 5- Implementazione in linguaggio APL

La realizzazione dell'algoritmo in APL si rivela interessante per il modo semplice e compatto con cui tale linguaggio consente di procedere nell'implementazione delle regole descritte nel paragrafo 2. Le caratteristiche dell'APL permettono di applicare simultaneamente ogni regola su tutta l'immagine, costruendo poi una struttura tridimensionale dove vengono allineate le matrici prodotte dall'L-sistema (cfr. Fig. 7).

L'implementazione in APL richiede che l'immagine sia "bordata" con un simbolo diverso da quello rispetto a cui si cercano le variazioni, per evitare perdite di informazione sui bordi. Qui, poiché si cercano le variazioni rispetto al simbolo 1, è stato scelto per la bordatura il simbolo 0.

La struttura che si ottiene dopo l'applicazione delle regole è una matrice tridimensionale di 8 piani, con righe e colonne uguali alla matrice contenente l'immagine dopo la bordatura, dove ogni piano è la matrice bidimensionale risultante dal confronto di tutti i punti dell'immagine con uno dei vicini. L'ordine dei piani rispetto all'ordine dei vicini è lo stesso della stringa di codifica (cfr. Fig. 5). La lettura della struttura lungo la dimensione dei piani da "così", per ogni punto di coordinate (x,y) appartenente all'immagine bordata, la stringa di codifica formata dai punti di coordinate (x,y) estratti da ciascun piano nell'ordine dato.

Fig. 8



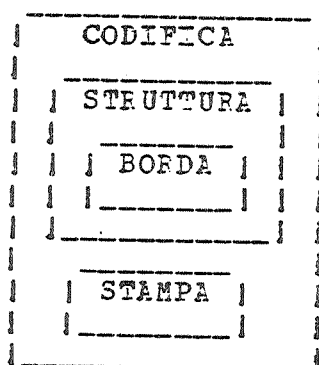
La stampa che viene prodotta su richiesta durante l'esecuzione dell'algoritmo mostra quanti sono i vicini con simbolo diverso rispetto ai punti con simbolo 1, proprio sommando lungo la direzione dei piani ed ottenendo così una prima semplice descrizione di questi punti di contorno.

L'uso dei simboli 0 ed 1 consente all'algoritmo implementato di sfruttare appieno la logica binaria su cui si basa il funzionamento della macchina, risparmiando notevoli quantità di memoria e di CPU. Inoltre, stabilendo una corrispondenza biunivoca secondo l'ordine illustrato in Fig. 5 tra l'insieme dei vicini e l'insieme dei bits di un byte, la stringa di codifica può essere fisicamente rappresentata da un solo byte.

Alla fine il programma fornisce una tabella che indica ad ogni riga le coordinate di un punto di contorno, seguite dalla stringa di codifica prodotta dall'L-sistema.

Il procedimento dell'algoritmo si articola in quattro funzioni APL (cfr. Lista delle funzioni APL). Il seguente schema ne illustra il concatenamento.

Fig. 9



BORDA (0)

La matrice 'A' data in ingresso viene bordata con '0' aggiungendo due righe e due colonne, e ruotando in modo che l'ultima riga e l'ultima colonna passino in prima posizione.

$$\begin{matrix} A \\ (2+pA)+A \\ -1\phi^{-1}e \end{matrix}$$

-----  
(0) Le istruzioni APL vengono interpretate procedendo da destra.

### STRUTTURA

La matrice 'A' bordata con '0' viene posta uguale ad 1 per trasformarla in effettiva matrice di bit; viene negata ed il risultato assegnato alla variabile 'N'.

1=BORDA A

Nelle due istruzioni viene costruita la struttura tridimensionale applicando le regole. L'applicazione e' realizzata confrontando via via ogni elemento con uno dei suoi otto vicini secondo un'istruzione del tipo:

N←~A

'A' viene cioe' congiunta con la sua negazione opportunamente ruotata. Il risultato del confronto con il vicino simmetrico e' concatenato secondo una terza dimensione (cfr. Fig. 8). Alla fine 'R' sara' un array di 8 piani, che sono le 8 matrici risultato dei confronti.

### STAMPA

L'istruzione: !IERSVPPF'[0 1 2 3 4 5 6 7 8]+[1]R]

costruisce una matrice di caratteri i cui elementi sono le lettere 'IERSVPPF0' indicanti se i vicini diversi da 1 sono rispettivamente 0 1 2 3 4 5 6 7 8.

### CODIFICA

Viene eseguita la funzione STRUTTURA. E' richiesto un input da terminale. Rispondendo 'SI' viene mandata in esecuzione la funzione STAMPA, che mostra i contorni dell'immagine secondo il

STRUTTURA A

'\*\*\*\*\*VUOI LA STAMPA?\*\*\*\*\*'  
→L1×11=^/'SI'≠↑□

STAMPA R



criterio sopra illustrato; altrimenti l'esecuzione prosegue con l'istruzione etichettata 'L1'. 'F' viene riordinata in una matrice di 8 colonne e di tante righe quanti sono gli elementi di 'A': su ogni riga stanno gli 8 elementi di coordinate (x,y) delle 8 matrici, vale a dire la stringa di codifica di un punto (cfr. Fig. 7 e 8). Alla variabile 'C' viene assegnato un vettore binario di dimensioni uguali al numero delle righe della nuova matrice: le posizioni degli 1 indicano le righe della matrice che contengono stringhe di codifica di punti di contorno. 'C' viene quindi usato per estrarre dalla matrice solo queste stringhe, che a loro volta formano una matrice bidimensionale assegnata alla variabile 'T' che contiene tutte le stringhe di codifica dei punti di contorno. Per associare a ciascuna riga di 'T' le coordinate del punto di cui essa e' la stringa di codifica, e' stato usato il seguente accorgimento. 'C', che ha tanti elementi quanti sono i punti dell'immagine dopo la bordatura, viene usata per formare un vettore di indici (1,2,...,n) delle stesse dimensioni di 'C' dove n e' appunto il numero complessivo dei suoi elementi. Con lo stesso metodo usato per estrarre le stringhe di codifica dei punti di contorno da 'R', si estraggono dal vettore degli indici quegli elementi che, decodificati in base (x,y) al numero delle righe e delle colonne di 'A',

$((x/1+pR),8)p,3 2 1QR$

$C+,v/[1]R$

$1pC$

$(1+pR)TC/1pC$

forniscono le coordinate dei punti di contorno nello stesso ordine delle stringhe di codifica in 'T'.

Così per es. se la matrice ha dimensioni  $10 \times 10$ , l'elemento 47 del vettore degli indici - che punta alla 47-esima riga di 'R', la quale è la stringa di codifica del 47-esimo punto dell'immagine fornisce, decodificandolo in base  $(10, 10)$  le coordinate del punto, cioè  $(4, 7)$ .

Bisogna poi aumentare di una unità gli indici delle righe, dato che decodificando per es. 3 in base  $(10, 10)$  si ottiene  $(0, 3)$ , mentre il terzo elemento sta già sulla prima riga.

$((pD)p1 0)+D$

La variabile 'TB' è l'uscita della funzione e contiene la tabella con le coordinate dei punti di contorno e le loro stringhe di codifica.

L'algoritmo è stato provato -sul sistema IBM 370/168 del CNUCE di Pisa che dispone di APL-CMS. Il tempo di CPU relativo ad un'applicazione su una matrice di 12.000 punti è risultato di due secondi.

Per verificare le possibilità e la correttezza del metodo proposto, è stato studiato infine un algoritmo successivamente tradotto in linguaggio ASSEMBLER/370. Tale algoritmo è stato verificato su matrici di diverse dimensioni ( $10 \times 10$ ,  $100 \times 100$ ,  $1000 \times 1000$ ) e in diverse condizioni di carico (nessun punto di contorno, massima concentrazione di punti di contorno). I risultati, in termini di CPU/TIME, ottenuti utilizzando il calcolatore IBM 370/168 sono risultati soddisfacenti: abbiamo infatti rilevato un tempo/punto di  $2.3$  microsecondi.

### Esempio 1

Nell'esempio sono presentati dieci possibili tipi di punto di contorno e la loro stringa di codifica seguita dalla lettura in base decimale della stringa (cfr. paragrafo 3).

STRINGA DI CODIFICA DEI PRIMI 10 TIPI DI PUNTO DI CONTORNO

1 1 1 0 1 1 1 1 1	0 0 0 0 0 0 0 1	1
0 1 1 1 1 1 1 1 1	0 0 0 0 0 0 1 1	2
0 1 1 0 1 1 1 1 1	0 0 0 0 0 0 1 1	3
1 0 1 1 1 1 1 1 1	0 0 0 0 0 1 0 0	4
1 0 1 0 1 1 1 1 1	0 0 0 0 0 1 0 1	5
0 0 1 1 1 1 1 1 1	0 0 0 0 0 1 1 0	6
0 0 1 0 1 1 1 1 1	0 0 0 0 0 1 1 1	7
1 1,0 1 1 1 1 1 1	0 0 0 0 1 0 0 0	8
1 1 0 0 1 1 1 1 1	0 0 0 0 1 0 0 1	9
0 1 0 1 1 1 1 1 1	0 0 0 0 1 0 1 1	10

Esempio\_2

L'esempio da' in successione il risultato dell'applicazione delle quattro regole di riscrittura sulla matrice 'A' dei dati. Esso corrisponde allo schema in Fig. 2.

Dopo l'applicazione di ogni regola e' mostrato con maggiore evidenza come ciascuna di esse agisce sull'immagine - nel nostro caso una casetta stilizzata.











Esempio 3

L'esempio mostra il risultato dell'algoritmo in API sulla casetta dell'esempio precedente.

La prima stampa è eseguita secondo i criteri già illustrati a pag. 11. Rispetto al risultato finale 'P' dell'esempio precedente, ora si nota una prima distinzione tra i diversi tipi di punto di contorno. Tutti i punti al vertice degli angoli retti interni sono contrassegnati da una 'I', gli esterni da una 'V'. con 'R' vengono indicati punti che appartengono ad una retta. In nona colonna si può vedere come è caratterizzato, un filamento indipendente che si distacca dal resto dell'immagine: i punti contrassegnati con 'S P V' indicano che i loro vicini con simbolo diverso sono rispettivamente 4 6 7.

L'interpretazione delle stringhe di codifica in base binaria consente di arricchire il numero dei simboli.

Nella seconda stampa della casetta si può osservare come i punti prima contrassegnati con 'R' sono rappresentati con 6 simboli diversi:

A	-	punto	di	retta	in	alto
B	-	"	"	"	"	basso
D	-	"	"	"	"	a destra
S	-	"	"	"	"	sinistra
E	-	"	"	"	"	obliqua a destra in alto
F	-	"	"	"	"	" sinistra in alto

Allo stesso modo si può procedere per gli angoli retti ed in generale per tutti i punti angolosi, essendo sempre dati otto casi diversi per ogni punto dello stesso tipo.

Nel nostro caso per tutti gli angoli retti interni abbiamo conservato il simbolo 'I', mentre gli esterni sono stati suddivisi in 'T' (punto di vertice di angolo retto esterno in alto) e 'V' (in basso).

I punti con simbolo 'W' hanno i loro vicini con simbolo diverso disposti senza continuità (secondo lo schema in fig. 5).

Con 'Q' sono indicati i vertici delle punte.

Situazioni interessanti sono la descrizione data del camino: retta in alto - angolo retto interno - retta a sinistra - angolo retto esterno in alto - retta in alto - angolo retto esterno in alto - retta a destra - angolo retto interno - retta in alto (i punti angolosi possono essere ulteriormente distinti in destri e sinistri); la descrizione della finestra con i punti di ponte centrali tutti identificati ed il contorno inferiore completamente distinto con punti di retta in alto da quello contiguo della casetta con punti di retta in basso. I contorni interni si distinguono inoltre per avere i simboli disposti inversamente rispetto ai contorni esterni.

La funzione 'DESCRIVI' è analoga a 'STAMPA' (cfr. p. 11).

### ESEMPIO DI APPLICAZIONE DELL'ALGORITMO

#### CASA

```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 0 0 0 0
0 0 0 0 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
0 0 0 0 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
0 0 0 0 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0

```

#### CODIFICA CASA

\*\*\*VUOI LA STAMPA?\*\*\*

SI

```

          VVV
          F F
SRRRRRRRRRRRRRRRRRI IRRRRPS
RI                       IR
RI                       IP
PSREI                    IRRRRRRI IRRS
E IIFII E S E E
R F S E E S E R
R R P R IRRRRRRI R
VRRV F VRRRRRRRRRRRRRV

```



8	6	1	0	0	0	0	0	0	1
8	9	0	0	1	0	0	0	0	0
8	10	0	1	0	0	0	0	0	0
8	11	1	0	1	0	0	0	0	0
8	12	0	1	0	0	0	0	0	0
8	13	1	0	0	0	0	0	0	0
8	18	0	0	1	1	0	0	0	0
8	22	1	0	1	1	0	0	0	1
8	26	1	0	0	0	0	0	0	1
8	29	0	0	1	1	0	0	0	0
9	6	1	0	0	0	0	0	1	1
9	9	0	0	1	1	0	0	0	0
9	11	1	0	1	1	0	0	0	1
9	13	1	0	0	0	0	0	0	1
9	18	0	0	0	1	1	0	0	0
9	22	0	0	0	1	1	0	1	1
9	26	0	0	0	0	0	0	1	1
9	29	0	0	1	1	1	0	0	0
10	6	1	0	0	0	0	0	1	1
10	9	0	0	1	1	1	0	0	0
10	11	1	0	1	1	1	0	1	1
10	13	1	0	0	0	0	0	1	1
10	18	0	0	0	0	1	0	0	0
10	19	0	0	0	0	1	1	0	0
10	20	0	0	0	0	1	1	1	0
10	21	0	0	0	0	0	1	1	0
10	22	0	0	0	0	1	0	1	0
10	23	0	0	0	0	1	1	0	0
10	24	0	0	0	0	1	1	1	0
10	25	0	0	0	0	0	1	1	0
10	26	0	0	0	0	0	0	1	0
10	29	0	0	1	1	1	0	0	0
11	6	1	1	1	0	0	0	1	1
11	7	1	1	1	0	0	0	0	0
11	8	1	1	1	0	0	0	0	0
11	9	1	1	1	1	1	0	0	0
11	11	1	1	1	1	1	0	1	1
11	13	1	1	1	0	0	0	1	1
11	14	1	1	1	0	0	0	0	0
11	15	1	1	1	0	0	0	0	0
11	16	1	1	1	0	0	0	0	0
11	17	1	1	1	0	0	0	0	0
11	18	1	1	1	0	0	0	0	0
11	19	1	1	1	0	0	0	0	0
11	20	1	1	1	0	0	0	0	0
11	21	1	1	1	0	0	0	0	0
11	22	1	1	1	0	0	0	0	0
11	23	1	1	1	0	0	0	0	0
11	24	1	1	1	0	0	0	0	0
11	25	1	1	1	0	0	0	0	0
11	26	1	1	1	0	0	0	0	0
11	27	1	1	1	0	0	0	0	0
11	28	1	1	1	0	0	0	0	0
11	29	1	1	1	1	1	0	0	0

DESCRIVI CASA

TAT  
S D  
RAAAAAAAAAAAAAAAAAAAT TAAAAAR  
FI IF  
FI TF  
PBBT TRPPWBBT TBB C  
S ITWIT D W S D  
S D W S D W S D  
S D W S TAAAWAAAT D  
VBBV W VBBBBBBBBBBBBBBV

Esempio 4

In questo esempio e' dato il risultato dell'applicazione dell'algoritmo su un'immagine reale. Si tratta della citta' di Pavia ottenuta da satellite e trattata con il software messo a punto presso il GAD-IFCTF (1).

La descrizione punto per punto che si ottiene della forma della citta', dove ogni punto assume un valore in corrispondenza al numero dei vicini (cfr. pag. 11), permette di osservare come anche su forme naturali rumorose l'algoritmo metta in risalto regolarita' e conformazioni caratteristiche.

Tutte le rette risultano identificate dalla lettera 'P' ripetuta almeno due volte. Le rientranze profonde sono descritte dalla lettera 'I' ripetuta piu' di una volta. Le forme rettangolari che si incontrano sul contorno - due a destra ed una a sinistra -, sono descritte con 'R' e, sui punti di vertice, 'V'. L'appendice che si stacca in basso a destra ha un andamento regolare, descritto dal succedersi delle lettere 'IR' ('FI'), 'IES' ('SEI') e 'IERS' ('SPFI') secondo la maggiore o minore inclinazione.

Una quantita' molto superiore di informazioni puo' essere ricavata da immagini reali rumorose quando, oltre che del numero, si tiene conto della posizione dei vicini. In questa direzione e' ora orientata la nostra ricerca.

姓名	性别	民族	出生年月	籍贯	文化程度	政治面貌	工作单位	职务
李德胜	男	汉族	1945.03	湖南长沙	高中	中共党员	长沙市教育局	科长
王小明	男	汉族	1950.07	湖北武汉	大学	中共党员	武汉市公安局	处长
张国强	男	汉族	1955.12	山东青岛	大学	中共党员	山东省公安厅	处长
刘为民	男	汉族	1960.05	河南郑州	大学	中共党员	河南省公安厅	处长
陈永年	男	汉族	1965.09	浙江杭州	大学	中共党员	浙江省公安厅	处长
赵子龙	男	汉族	1970.02	四川成都	大学	中共党员	四川省公安厅	处长
孙文杰	男	汉族	1975.06	广东广州	大学	中共党员	广东省公安厅	处长
周大伟	男	汉族	1980.10	福建福州	大学	中共党员	福建省公安厅	处长
吴昊	男	汉族	1985.04	江西九江	大学	中共党员	江西省公安厅	处长
郑宇	男	汉族	1990.08	广西柳州	大学	中共党员	广西壮族自治区公安厅	处长
李德胜	男	汉族	1945.03	湖南长沙	高中	中共党员	长沙市教育局	科长
王小明	男	汉族	1950.07	湖北武汉	大学	中共党员	武汉市公安局	处长
张国强	男	汉族	1955.12	山东青岛	大学	中共党员	山东省公安厅	处长
刘为民	男	汉族	1960.05	河南郑州	大学	中共党员	河南省公安厅	处长
陈永年	男	汉族	1965.09	浙江杭州	大学	中共党员	浙江省公安厅	处长
赵子龙	男	汉族	1970.02	四川成都	大学	中共党员	四川省公安厅	处长
孙文杰	男	汉族	1975.06	广东广州	大学	中共党员	广东省公安厅	处长
周大伟	男	汉族	1980.10	福建福州	大学	中共党员	福建省公安厅	处长
吴昊	男	汉族	1985.04	江西九江	大学	中共党员	江西省公安厅	处长
郑宇	男	汉族	1990.08	广西柳州	大学	中共党员	广西壮族自治区公安厅	处长



```

          SRS
        SI IS
        R E
    VRRS R IERV R IS
      R E R R
    SI E SRS E R
      E III IR SEI R
    SI III E IS
      R E VRRV
      E R R
    VREI IES RERV
      R IR SREE
      R III IERRRS
      R IES
      R IS
      R E
      R IS
      R E
      R IERRRRRRRS
      R IERS
      R IEV
      R R
      R E
      R IERS
      R IS
      R E
      R IERRRRV
      R
      R IERRRRV
      R E
      R IEV R IERV
      R VRRRRRS
      R
      R IIEEV
      R I R
      R IIR
      R IERRRS
      R IR
      R IR
      R SEI IR
      R SEI IES
      R RI IES
      R FI IERS
      R SEI IES
      R SREI IES
      R SEI IES
      RRRRRRSVE
    
```

B I B L I O G R A F I A

- (1) F. MASSIMO, P. MUSSIO, Strumentazione sperimentale software per trattamento di dati ottenuti da rilievi termografici, Atti del 'Seminar on the lacustrine environment', Como Novembre 1975.
- (2) K. S. FU, P. N. SWAIN, On syntactic pattern recognition, in Software Engineering, vol. II, New York 1971.  
  
P. MUSSIO ed altri, A method for the automatic analysis of gamma-ray events in astronomical spark chamber, Computer Physics Communication 4-1972 299-314.
- (3) G.T. HERMAN, G. ROZENBERG, Developmental system and languages, North Holland 1975.
- (4) A. DELLA VENTURA, G. FRESTA, P. MUSSIO, A fully automated scanning system for astronomical spark chamber images, Atti del 'Conference on Computer Assisted Scanning, Padova 21-24 Aprile 1976.  
  
idem, A morphological RW system for gamma-astronomy, GAD 1975.
- (5) F. MASSIMO ed altri, Un metodo per l'estrazione la codifica ed il trattamento dei contorni, GAD 1976.

MAINI-MERAVIGLIA-MUSSIO, introduzione al sistema formale per la composizione di immagini, Rapporto Interno LFCTP, GAD 1974.

CUGINI-CAVAGNA-MERAVIGLIA-MUSSIO, Consideration on two implemetnations of an image generation and recognition system, GAD 1975.

LISTA DELLE FUNZIONI APL

VCODIFICA[ ]V

```
V TB←CODIFICA A;C;D;R;T
[1] R←STRUTTURA A
[2] '****VUOI LA STAMPA?****'
[3] →L1 ×11=Λ/'SI'≠2+[]
[4] STAMPA R
[5] L1:T←(C←,v/[1]R)/[1]((×/1+ρR),8)ρ,3 1 2QR
[6] TB←(((ρD)ρ1 0)+D+Q(1+ρR)TC/1ρC),T
V
```

VSTRUTTURA[ ]V

```
V R←STRUTTURA A;N
[1] R←(AΛ1Φ-1ΘN),[1](AΛ-1ΘN),[1](AΛ-1Φ-1ΘN),[0.5]AΛ-1ΦN←A←1=BORDA A
[2] R←((AΛ-1Φ1ΘN),[1](AΛ1ΘN),[1](AΛ1Φ1ΘN),[0.5]AΛ1ΦN),[1]R
V
```

VBORDA[ ]V

```
V M←BORDA A
[1] M←-1Φ-1Θ(2+ρA)+A
V
```

VSTAMPA[ ]V

```
V STAMPA R
[1] IERSVP FO'[0 1 2 3 4 5 6 7 8]+/[1]R
V
```