
La Matematica nella Società e nella Cultura

RIVISTA DELL'UNIONE MATEMATICA ITALIANA

FRANCESCO LEONETTI, MANUELA MANETTA

Autosimilarità e trattamento delle immagini

La Matematica nella Società e nella Cultura. Rivista dell'Unione Matematica Italiana, Serie 1, Vol. 8 (2015), n.2, p. 225–278.

Unione Matematica Italiana

http://www.bdim.eu/item?id=RIUMI_2015_1_8_2_225_0

L'utilizzo e la stampa di questo documento digitale è consentito liberamente per motivi di ricerca e studio. Non è consentito l'utilizzo dello stesso per motivi commerciali. Tutte le copie di questo documento devono riportare questo avvertimento.

*Articolo digitalizzato nel quadro del programma
bdim (Biblioteca Digitale Italiana di Matematica)
SIMAI & UMI*

<http://www.bdim.eu/>

La Matematica nella Società e nella Cultura. Rivista dell'Unione Matematica Italiana, Unione Matematica Italiana, 2015.

Autosimilarità e trattamento delle immagini

FRANCESCO LEONETTI - MANUELA MANETTA

Da alcuni decenni ormai, l'uso del calcolatore è sempre più diffuso per le più svariate applicazioni. Tale strumento, come tutti sappiamo, dispone di uno spazio di memoria limitato, così come accade per i Compact Disc, Digital Versatile Disc, Hard Disk esterni e Pen Drive. Allo stesso tempo ognuno di noi avverte la necessità di memorizzare su questi dispositivi il maggior numero possibile di file (file di testo, immagini, software). Per far sì che tali file occupino uno spazio di memoria ridotto, si usano metodi di compressione (chi di noi non si è mai imbattuto in file .zip o .rar?).

Nel seguito ci occuperemo della cosiddetta compressione frattale di immagini, mostrando come le idee che la ispirano si fondano su alcuni risultati di matematica. Essa risale ad una ventina di anni fa, [1], [3] e si fonda sulla constatazione che alcune figure geometriche molto frastagliate (dette frattali) possono essere approssimate mediante un procedimento iterativo basato sul *Teorema delle contrazioni*. Tale teorema, studiato per lo più nell'ambito dell'analisi matematica, si applica dunque in diverse branche della matematica e dell'informatica; costituisce infatti la base del discorso sulla compressione di immagini.

TEOREMA 1 (delle contrazioni). – *Sia X uno spazio metrico completo e sia $G : X \rightarrow X$ una contrazione, allora esiste uno ed un solo $x^* \in X$ tale che $G(x^*) = x^*$. Inoltre, per ogni $a \in X$, posto $x_1 = a$, $x_2 = G(x_1)$, $x_3 = G(x_2)$, ..., $x_{n+1} = G(x_n)$, risulta che x_n converge all'elemento x^* di cui sopra.*

Per poter applicare tale teorema alle immagini, abbiamo bisogno di rappresentarle matematicamente come insiemi compatti del piano \mathbb{R}^2 e definire uno spazio ad esse relativo a cui applicare il teorema, lo spazio X degli insiemi compatti non vuoti di \mathbb{R}^2 . Abbiamo bisogno,

inoltre, di una metrica che definisca le distanze tra gli oggetti di questo spazio: la metrica di Hausdorff.

Considereremo due fasi: la fase di compressione dell'immagine, in cui si memorizzano le informazioni necessarie sulla figura, e la fase di decompressione, quella attraverso cui sarà possibile ricostruire l'immagine di partenza. Nella prima parte del teorema, ci si chiede se una figura i possa essere decomposta in un numero finito di copie rimpicciolite di se stessa (immagine autosimile); in tal caso è possibile avere una contrazione G in modo che la nostra immagine i sia punto fisso di G ; sia x^* la figura che si vuole approssimare e sia G una contrazione opportuna che abbia x^* come punto fisso; tale contrazione sarà realizzata mediante omotetie, traslazioni e rotazioni, [8], [9]. Il teorema precedente assicura che, dopo aver fissato la G , la figura del piano x^* è l'unica ad avere la proprietà di punto fisso. Dunque la figura x^* risulta univocamente determinata dalla contrazione G . Questo vuol dire che per memorizzare la figura x^* basta memorizzare la contrazione G : ciò produce un risparmio di memoria se i parametri con cui si memorizza G sono in numero inferiore rispetto ai punti che servono per disegnare x^* . Nella fase di decompressione ci si preoccupa di come ottenere x^* a partire dalla contrazione G . A questo proposito si sfrutta la seconda parte del teorema delle contrazioni: si prende una figura a piacere a , la si trasforma con G , il risultato viene trasformato a sua volta con G e così di seguito. Dopo un certo numero di iterazioni si giunge ad un'approssimazione della figura x^* .

La proprietà fondamentale per la costruzione degli operatori che permettono la compressione e la ricostruzione dell'immagine è l'auto-similarità. Il triangolo di Sierpinski è autosimile, come pure un quadrato pieno. Non tutte le immagini, però, sono autosimili: ad esempio la circonferenza e il cerchio non godono di tale proprietà. Tra le figure non autosimili ve ne sono alcune che hanno la proprietà di essere localmente autosimili, ovvero sono decomponibili in copie rimpicciolite di pezzi della figura stessa, come accade per esempio per una "V" o una casetta.

Come si applica ad esse il metodo di compressione?

Finora si sono prese in considerazione similitudini contrattive; cosa accade alle immagini se si lasciano agire su di esse funzioni lip-schitziane che non siano necessariamente similitudini?

Cercheremo, nel seguito, di dare una risposta a tali quesiti.

2. – Autosimilarità

Si consideri una figura i . Ci si chiede se sia possibile decomporre i in k copie rimpicciolite di se stessa, cioè

$$(1) \quad i = W_1(i) \cup W_2(i) \cup \dots \cup W_k(i)$$

dove W_1, W_2, \dots, W_k sono similitudini contrattive. Se si riesce a farlo (con k non molto grande), allora la compressione di i sarà la memorizzazione dei parametri di W_1, W_2, \dots, W_k . La decompressione sarà fatta mediante il teorema delle contrazioni: si definisce l'operatore G ,

$$G(a) = W_1(a) \cup W_2(a) \cup \dots \cup W_k(a)$$

e si parte da una qualunque figura a , successivamente si trasforma a con G creando la successione

$$x_1 = a, x_2 = G(x_1), x_3 = G(x_2), \dots$$

Il teorema delle contrazioni assicurerà che tale successione approssimerà la figura di partenza i perché essa è, per costruzione, la figura fissa della contrazione G , infatti

$$i = W_1(i) \cup W_2(i) \cup \dots \cup W_k(i) = G(i).$$

La possibilità di decomporre una figura in un numero finito di copie rimpicciolite di se stessa va sotto il nome di autosimilarità. Tale possibilità vale per tutte le figure?

Esempi di figure autosimili sono:

- il *triangolo di Sierpinski*:

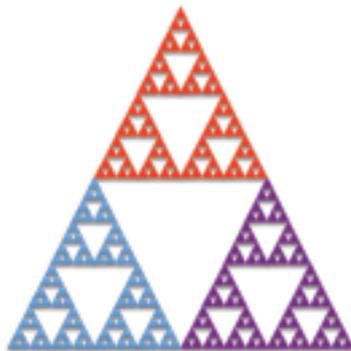


Figura 1: autosimilarità del triangolo di Sierpinski.

il pezzo celeste a sinistra, il pezzo viola a destra e il pezzo rosso in alto sono le tre copie rimpicciolite in cui si decompone il triangolo di Sierpinski;

- il *quadrato*:

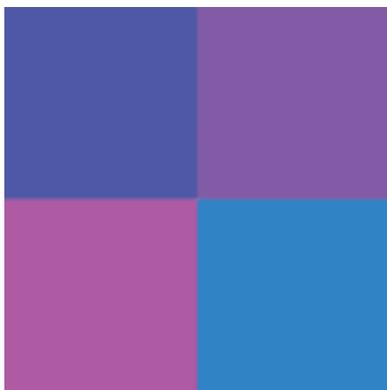


Figura 2: autosimilarità del quadrato.

per verificare che si tratta di una figura autosimile basta congiungere i punti medi dei lati opposti del quadrato e notare che si ottengono quattro quadrati che hanno per lato la metà del lato della figura di partenza e sono opportunamente traslati in alto e/o a destra di mezzo lato.

È dunque possibile decomporre il quadrato in quattro copie rimpicciolite di se stesso.

In simboli, sia $W_j(i)$ ogni copia rimpicciolita con $j = 1, 2, 3, 4$,

$$i = W_1(i) \cup W_2(i) \cup W_3(i) \cup W_4(i).$$

La proprietà di autosimilarità non è una peculiarità dei frattali. Infatti, nell'esempio appena proposto, il quadrato è un'immagine autosimile, ma non un frattale.

Pensiamo che l'aggettivo "frattale" sia stato assegnato a questo tipo di compressione di immagini quando il teorema delle contrazioni fu utilizzato per l'approssimazione di frattali. È da notare, però, che il metodo si basa sulla proprietà di autosimilarità.

Per verificare l'autosimilarità di altre figure, come la circonferenza, lo smile, il cerchio, bisogna introdurre alcune nozioni riguardanti le proprietà delle *similitudini*.

Una *similitudine* è una trasformazione del piano in sé che conserva i rapporti fra le distanze; precisamente, $S : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ è una similitudine se esiste $\rho \in (0, +\infty)$ tale che, per ogni scelta di $x, y \in \mathbb{R}^2$ si ha

$$(2) \quad d(S(x), S(y)) = \rho d(x, y)$$

La costante di proporzionalità ρ è detta costante di Lipschitz. Se ρ è minore di uno i punti vengono “avvicinati” dalla trasformazione (in questo caso si parla di *contrazione*); se, al contrario, la costante è maggiore di uno i punti sono “allontanati” mediante la similitudine. Ad esempio, le omotetie, le traslazioni e le rotazioni sono similitudini.

Ricordiamo che l'*omotetia di fattore* $\lambda \in (0, +\infty)$ è $O_\lambda : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ così definita: $O_\lambda(x) = \lambda x$; essa contrae o dilata a seconda che la costante λ sia minore o maggiore di 1.

Fissato un vettore $w \in \mathbb{R}^2$, la *traslazione* $T_w : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ è definita da $T_w(x) = x + w$; essa ha l'effetto di muovere i punti nella stessa direzione mantenendo fisse le distanze fra essi.

Una *trasformazione ortonormale*, *Ort*, da \mathbb{R}^2 in \mathbb{R}^2 è un'isometria lineare. Ad esempio una rotazione è una trasformazione ortonormale.

Osserviamo che traslazioni e rotazioni sono particolari similitudini con costante di Lipschitz pari ad 1, mentre un'omotetia di fattore λ è una similitudine con costante di Lipschitz pari a λ .

È immediato verificare che una similitudine è iniettiva; è più laborioso, invece, mostrare che sia suriettiva.

Per fare ciò, è facile vedere che le omotetie e le traslazioni sono suriettive. Le trasformazioni ortonormali sono lineari e iniettive quindi sono suriettive per il teorema di isomorfismo. Una generica similitudine può essere fattorizzata mediante una omotetia, una traslazione e una trasformazione ortonormale (si veda [8]) quindi è suriettiva perchè composta di applicazioni suriettive.

Osserviamo che una similitudine S di fattore ρ trasforma la circonferenza di centro x_0 e raggio r nella circonferenza di centro $S(x_0)$ e raggio ρr .

Se $\rho < 1$ allora la circonferenza trasformata ha un raggio più piccolo della circonferenza di partenza, quindi tale circonferenza trasformata

interseca quella di partenza al più in due punti. Quest'ultimo fatto è la chiave per mostrare che la circonferenza non è autosimile.

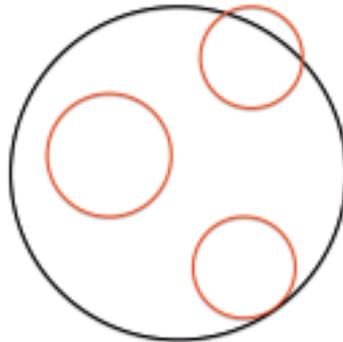


Figura 3: Non autosimilarità della circonferenza.

TEOREMA 2. – *La circonferenza non è autosimile*

Dimostrazione: [10] Per assurdo, se la circonferenza fosse autosimile, varrebbe la decomposizione (1) per opportune similitudini contrattive W_1, W_2, \dots, W_k . Per quanto detto prima, $W_1(i)$ è una circonferenza di raggio minore di quella di partenza i , dunque $W_1(i)$ ha in comune con i al più due punti di i ; le k copie rimpicciolite $W_1(i), W_2(i), \dots, W_k(i)$ ricostruiranno al più $2k$ punti di i ; i , però, ha infiniti punti; questo contraddice l'uguaglianza (1). \square

Si consideri ora la figura 4 che rappresenta un faccino sorridente, comunemente detto smile.

Tale figura è composta da tre circonferenze (una grande per raffigurare il contorno del viso e due piccole per gli occhi) e da un arco di circonferenza (per la bocca). Sfruttando lo stesso argomento utilizzato per il teorema precedente (si veda [11]) si prova il seguente

TEOREMA 3. – *Lo smile non è autosimile*

Cosa si può dire, invece, sull'autosimilarità del cerchio?

Per prima cosa osserviamo che, come per il caso della circonferenza, una similitudine S con costante di Lipschitz ρ , trasforma il cerchio di centro x_0 e raggio r nel cerchio di centro $S(x_0)$ e raggio ρr .



Figura 4: Smile.

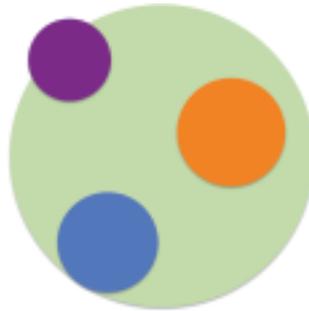


Figura 5: Non autosimilarità del cerchio

TEOREMA 4. – *Il cerchio non è autosimile*

Dimostrazione: [11] Per assurdo, se il cerchio i fosse autosimile, varrebbe la decomposizione (1) per opportune similitudini contrattive W_1, W_2, \dots, W_k . Per quanto detto prima, $W_1(i)$ è un cerchio di raggio minore di quello di partenza. Esso può avere tre collocazioni rispetto al cerchio di partenza i : si trova tutto all'interno di i , interseca il bordo di i in più punti o è tangente al bordo di i . Il primo caso è chiaramente accettabile, ma non contribuisce alla ricostruzione del bordo del cerchio i . Il secondo non è accettabile in quanto si uscirebbe dal cerchio di partenza i , cosa proibita dall'uguaglianza (1). Il terzo caso è accettabile, ma una copia rimpicciolita che sia tangente al bordo, ha un solo punto in comune con esso, che risulta quindi impossibile da ricostruire con un numero finito di copie rimpicciolite del cerchio. \square

Consideriamo, ora, la figura che rappresenta una “V”.

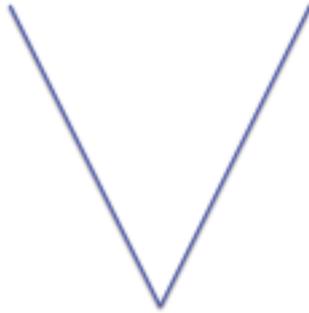


Figura 6: V

Essa può essere descritta, in termini matematici, come l’unione di due segmenti che hanno un punto in comune e che formano un angolo più piccolo dell’angolo piatto. Vogliamo dimostrare che non è autosimile.

TEOREMA 5. – *La figura V non è autosimile*

Dimostrazione: [11] Per assurdo, se la V fosse autosimile, varrebbe la decomposizione (1) per opportune similitudini contrattive. Questo vorrebbe dire che ogni copia rimpicciolita della V dovrebbe essere contenuta nella figura di partenza e quest’ultima dovrebbe essere completamente ricoperta dalle sue copie rimpicciolite.

Ricordiamo che una similitudine trasforma un segmento in un segmento e conserva gli angoli. Questo vuol dire che ogni copia rimpicciolita è una V più piccola ma con lo stesso angolo della V di partenza. Può il vertice della copia rimpicciolita non coincidere col vertice della figura di partenza? No, perché la decomposizione (1) forza le copie rimpicciolite ad essere contenute dentro la figura di partenza (figura 7). Allora ognuna delle copie rimpicciolite lascia fuori i due estremi della V. Dunque la (1) non può valere. Questo vuol dire che la V non è autosimile. \square



Figura 7: Caso possibile e caso impossibile.

In generale, davanti ad un'immagine, potrebbe non essere evidente se essa goda della proprietà di autosimilarità, quindi come sia possibile applicare il metodo di compressione all'immagine in questione. Ci viene in aiuto il seguente teorema del collage.

TEOREMA 6 (Teorema del Collage, [1] pag 100). – *Consideriamo una figura T e W_1, \dots, W_N similitudini contrattive con fattore di contrazione $0 < s < 1$, tali che il collage dei pezzi ottenuti trasformando T mediante le similitudini sia vicino alla figura di partenza, cioè*

$$h\left(T, \bigcup_{n=1}^N W_n(T)\right) \leq \varepsilon$$

con ε sufficientemente piccolo, dove h è la distanza di Hausdorff tra la figura T e la figura $\bigcup_{n=1}^N W_n(T)$. Le contrazioni $\{W_1, \dots, W_N\}$ creano in modo univoco l'operatore G che associa ad ogni figura F la figura $\bigcup_{n=1}^N W_n(F)$. Tale operatore G risulta essere una contrazione di fattore s dallo spazio X di tutte le figure in X stesso. Allora il Teorema delle contrazioni garantisce l'esistenza di un'unica figura fissa A per G .

Risulta

$$h(T, A) \leq \frac{\varepsilon}{1 - s}.$$

Data una figura T , il teorema del collage necessita le similitudini W_1, \dots, W_N tali che il collage $\bigcup_{n=1}^N W_n(T)$ sia vicino alla figura di partenza T ; la memorizzazione di W_1, \dots, W_N costituirà la com-

pressione di T . Trovare tali similitudini sembra richiedere una certa forma di interazione con l'utente, il quale si trova a dover manualmente operare la scelta delle trasformazioni opportune (si veda [1] pag 101 e 173). Nel capitolo 6 di [1] viene mostrato un modo automatico per ottenere la compressione di T . Esso si fonda sull'indebolimento della proprietà di autosimilarità, come illustrato nel seguente paragrafo.

2. – Locale autosimilarità

Abbiamo visto che quando una figura è decomponibile in copie rimpicciolite di se stessa essa si dice autosimile.

Quando una figura è decomponibile in copie rimpicciolite di pezzi della figura stessa essa si dice *localmente autosimile*:

$$(3) \quad i = W_1(i \cap R_1) \cup W_2(i \cap R_2) \cup \dots \cup W_k(i \cap R_k)$$

dove W_1, W_2, \dots, W_k sono similitudini contrattive e $W_j(i \cap R_j)$ è una copia rimpicciolita non di tutta l'immagine, ma solo del pezzo che cade dentro all'insieme R_j .

Abbiamo mostrato che la V non è autosimile; però essa è localmente autosimile. Verifichiamo tale proprietà in maniera costruttiva. Consideriamo i rettangoli R_1 ed R_2 come in figura 8. Ora consideriamo separatamente la parte di figura che cade in R_1 e quella che cade in R_2 ; riduciamo della metà entrambe le parti. È possibile ricostruire la figura di partenza mediante le copie rimpicciolite dei due pezzi della V appena ottenute semplicemente trasladole opportunamente. Si ha la decomposizione $i = W_1(i \cap R_1) \cup W_2(i \cap R_2) \cup W_3(i \cap R_1) \cup W_4(i \cap R_2)$, illustrata nella figura 9.

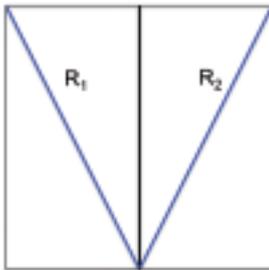


Figura 8: la figura V nei rettangoli R_1 ed R_2 .

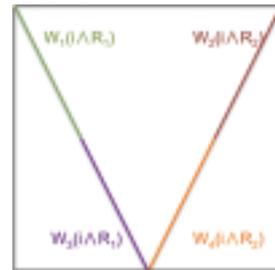


Figura 9: Decomposizione della V .

3. – Compressione e ricostruzione di un'immagine

Per la figura V la scomposizione in pezzi è stata fatta da noi tenendo conto delle caratteristiche della figura in esame. Volendo creare un algoritmo di decomposizione che agisca su un'immagine generica, dobbiamo trovare un modo universale per decomporre tale immagine in pezzi in modo che siano "approssimativamente" copie rimpicciolite di altri pezzi della figura di partenza. Come detto in precedenza, vogliamo costruire un operatore che agisca su sottoinsiemi della figura e che li confronti con l'immagine di partenza utilizzando la distanza di Hausdorff.

3.1 – Esempio 1

Descriviamo questo metodo universale ancora sulla V.

3.1.1 – Processo di compressione dell'immagine

Raffiguriamo la nostra immagine i in un quadrato 64×64 pixel. Facciamo due copie di i : dividiamo quella di sinistra in quadrati 16×16 pixel (sarà quindi una griglia formata da 16 quadrati, D_i con $i = 1, \dots, 16$) e quella di destra in quadrati 32×32 (griglia di 4 quadrati, R_j con $j = 1, \dots, 4$).

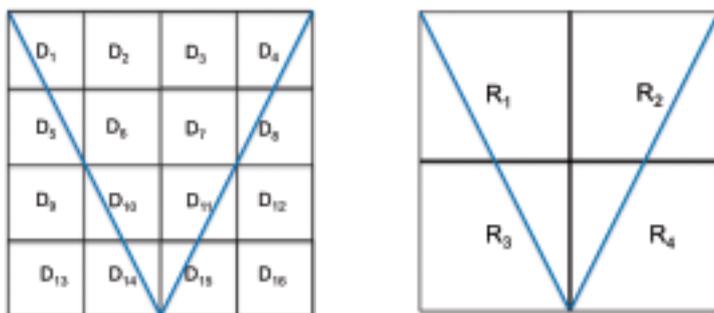


Figura 10: griglia D e griglia R.

Consideriamo ora il frammento di immagine che cade in R_1 e rimpiccioliamolo della metà. Operiamo allo stesso modo con R_2 , R_3 ed R_4 : in figura 11 otteniamo così le copie rimpicciolite dei quattro pezzi della figura i .

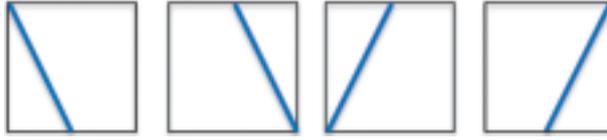


Figura 11: copie rimpicciolite di pezzi della V

Confrontando ora le immagini così ottenute con ciascun D_i , otteniamo le seguenti associazioni:

$$D_1 \leftarrow R_1$$

$$D_4 \leftarrow R_2$$

$$D_5 \leftarrow R_3$$

$$D_8 \leftarrow R_4$$

$$D_{10} \leftarrow R_1$$

$$D_{11} \leftarrow R_2$$

$$D_{14} \leftarrow R_3$$

$$D_{15} \leftarrow R_4$$

Inizialmente avevamo bisogno di $64 \times 64 = 4096$ informazioni, una per ogni pixel (acceso o spento); con le associazioni abbiamo ridotto ad 8 le informazioni necessarie per la memorizzazione della figura. La creazione di questa lista è il cosiddetto processo di compressione dell'immagine.

3.1.2 – Processo di ricostruzione dell'immagine

Bisogna ora definire l'operatore che ci permetterà di ricostruire l'immagine compressa a partire dalle associazioni sopra elencate:

$$L(a) = W_1(a \cap R_1) \cup W_4(a \cap R_2) \cup W_5(a \cap R_3) \cup W_8(a \cap R_4) \\ \cup W_{10}(a \cap R_1) \cup W_{11}(a \cap R_2) \cup W_{14}(a \cap R_3) \cup W_{15}(a \cap R_4)$$

dove $W_k(a \cap R_j)$ è l'insieme che si ottiene prendendo il pezzo di a che cade in R_j , riducendolo con l'omotetia di fattore $1/2$ e trasladolo in D_k . Applichiamo dunque l'operatore trovato L alla seguente figura a : il quadrato tutto pieno.



Per ottenere $L(a)$ si deve suddividere a con la griglia dei quadrati 32×32 :



Inoltre bisogna considerare la griglia dei quadrati 16×16 che è vuota:

D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

essa sarà riempita opportunamente mediante l'azione dell'operatore L .

$W_1(a \cap R_1)$: D_1 viene riempito con la parte di a che cade in R_1

D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

$W_4(a \cap R_2)$: D_4 viene riempito con la parte di a che cade in R_2

D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

$W_5(a \cap R_3)$: D_5 viene riempito con la parte di a che cade in R_3

D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

$W_8(a \cap R_4)$: D_8 viene riempito con la parte di a che cade in R_4

D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

$W_{10}(a \cap R_1)$: D_{10} viene riempito con la parte di a che cade in R_1

D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

$W_{11}(a \cap R_2)$: D_{11} viene riempito con la parte di a che cade in R_2

D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

$W_{14}(a \cap R_3)$: D_{14} viene riempito con la parte di a che cade in R_3

D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

$W_{15}(a \cap R_4)$: D_{15} viene riempito con la parte di a che cade in R_4

D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

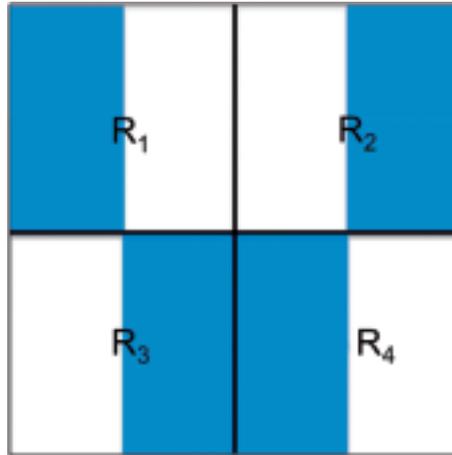
Ecco quindi la prima iterazione, ovvero $L(a)$:



Figura 12: $L(a)$.

Applichiamo ora L ad $L(a)$.

Per ottenere $L(L(a))$ bisogna suddividere $L(a)$ con la griglia dei quadrati 32×32 :



Inoltre si deve considerare la griglia dei quadrati 16×16 che è vuota; essa sarà riempita opportunamente mediante l'azione dell'operatore L .

D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

Anche in questo caso possiamo mostrare passo per passo le operazioni che L compie sulla figura $L(a)$.

$W_1(L(a) \cap R_1)$. D_1 viene riempito con la parte di $L(a)$ che cade in R_1

D ₁	D ₂	D ₃	D ₄
D ₅	D ₆	D ₇	D ₈
D ₉	D ₁₀	D ₁₁	D ₁₂
D ₁₃	D ₁₄	D ₁₅	D ₁₆

$W_4(L(a) \cap R_2)$. D_4 viene riempito con la parte di $L(a)$ che cade in R_2

D ₁	D ₂	D ₃	D ₄
D ₅	D ₆	D ₇	D ₈
D ₉	D ₁₀	D ₁₁	D ₁₂
D ₁₃	D ₁₄	D ₁₅	D ₁₆

$W_5(L(a) \cap R_3)$. D_5 viene riempito con la parte di $L(a)$ che cade in R_3

D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

$W_8(L(a) \cap R_4)$. D_8 viene riempito con la parte di $L(a)$ che cade in R_4 .

D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

$W_{10}(L(a) \cap R_1)$. D_{10} viene riempito con la parte di $L(a)$ che cade in R_1

D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

$W_{11}(L(a) \cap R_2)$. D_{11} viene riempito con la parte di $L(a)$ che cade in R_2

D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

$W_{14}(L(a) \cap R_3)$. D_{14} viene riempito con la parte di $L(a)$ che cade in R_3

D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

$W_{15}(L(a) \cap R_4)$. D_{15} viene riempito con la parte di $L(a)$ che cade in R_4

D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

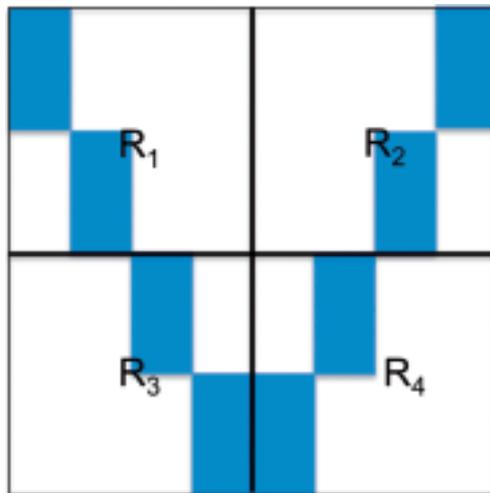
Ecco quindi il risultato della seconda iterazione, ovvero $L(L(a))$:



Figura 13: $L(L(a))$.

Applichiamo ora L ad $L(L(a))$.

Per ottenere $L(L(L(a)))$ si deve suddividere $L(L(a))$ con la griglia dei quadrati 32×32 :



Inoltre bisogna considerare la griglia dei quadrati 16×16 che è vuota; essa sarà riempita opportunamente mediante l'azione dell'operatore L .

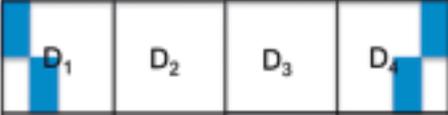
D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

Di nuovo mostriamo passo per passo le operazioni che L compie sulla figura $L(L(a))$.

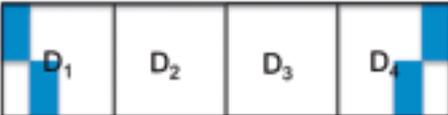
$W_1(L(L(a)) \cap R_1)$. D_1 viene riempito con la parte di $L(L(a))$ che cade in R_1

 D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

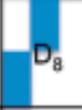
$W_4(L(L(a)) \cap R_2)$. D_4 viene riempito con la parte di $L(L(a))$ che cade in R_2

 D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

$W_5(L(L(a)) \cap R_3)$. D_5 viene riempito con la parte di $L(L(a))$ che cade in R_3

 D_1	D_2	D_3	D_4
D_5	D_6	D_7	D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

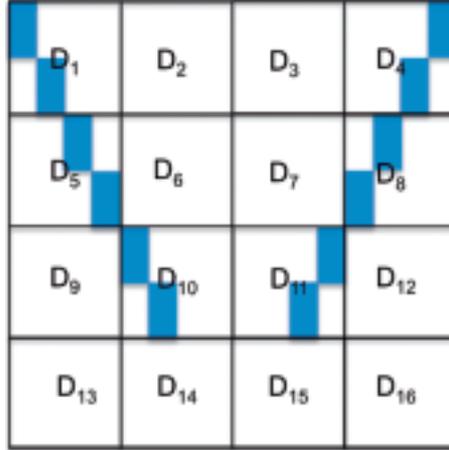
$W_8(L(L(a)) \cap R_4)$. D_8 viene riempito con la parte di $L(L(a))$ che cade in R_4

 D_1	D_2	D_3	D_4 
D_5 	D_6	D_7	 D_8
D_9	D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

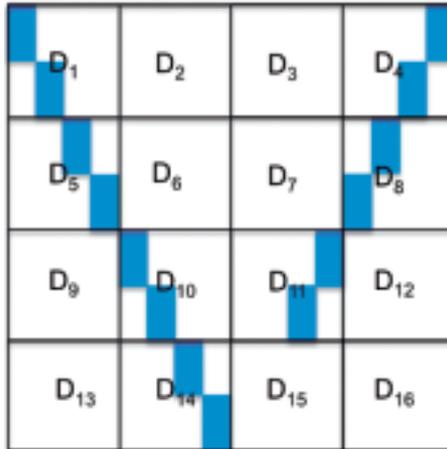
$W_{10}(L(L(a)) \cap R_1)$. D_{10} viene riempito con la parte di $L(L(a))$ che cade in R_1

 D_1	D_2	D_3	D_4 
D_5 	D_6	D_7	 D_8
D_9	 D_{10}	D_{11}	D_{12}
D_{13}	D_{14}	D_{15}	D_{16}

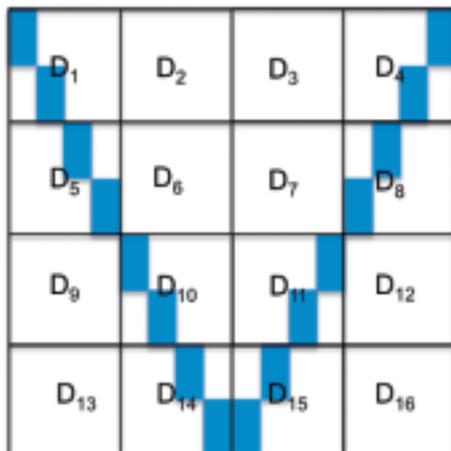
$W_{11}(L(L(a)) \cap R_2)$. D_{11} viene riempito con la parte di $L(L(a))$ che cade in R_2



$W_{14}(L(L(a)) \cap R_3)$. D_{14} viene riempito con la parte di $L(L(a))$ che cade in R_3



$W_{15}(L(L(a)) \cap R_4)$. D_{15} viene riempito con la parte di $L(L(a))$ che cade in R_4



Ecco quindi il risultato, ovvero $L(L(L(a)))$:

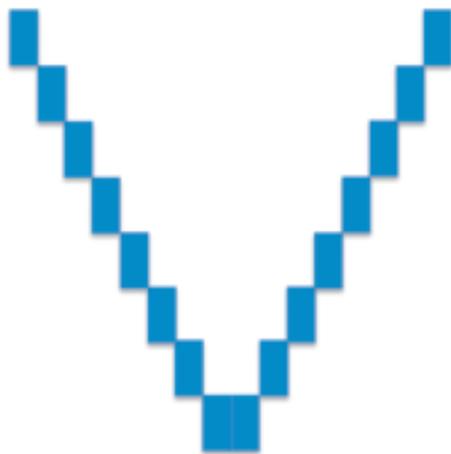
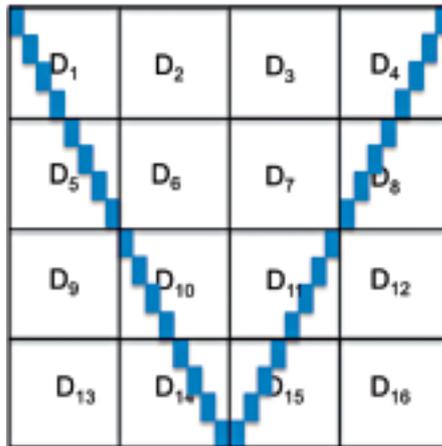
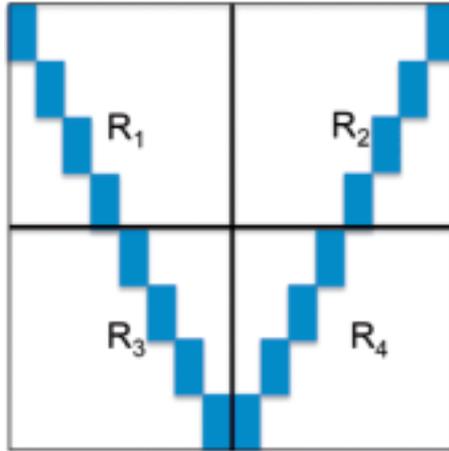


Figura 14: $L(L(L(a)))$.

Applicando di nuovo l'operatore alla figura ottenuta si ha:



La figura ottenuta dopo 5 iterazioni è la seguente:



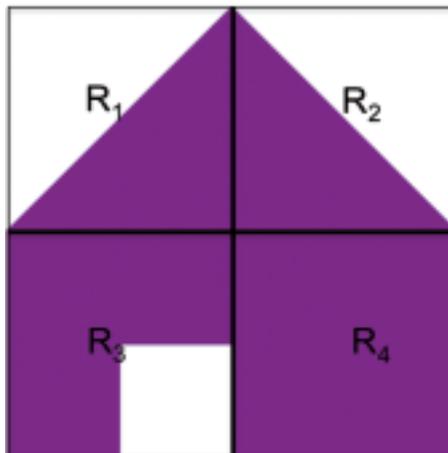
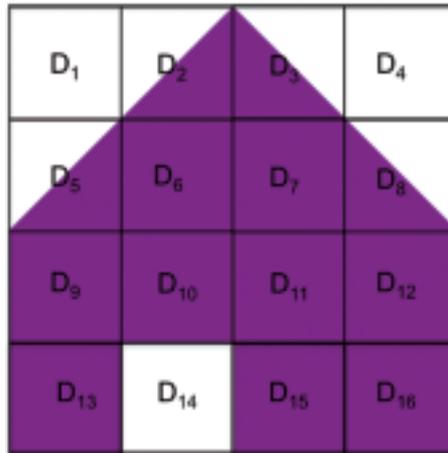
3.2 – Esempio 2

Consideriamo ora l'esempio di una casetta:



3.2.1 – Processo di compressione dell'immagine

Procedendo come nell'esempio precedente, cioè costruendo le griglie 16×16 e 32×32 per la casetta, come nelle seguenti figure



troviamo le seguenti associazioni:

$$D_2 \leftarrow R_1$$

$$D_3 \leftarrow R_2$$

$$D_5 \leftarrow R_1$$

$$D_6 \leftarrow R_4$$

$$D_7 \leftarrow R_4$$

$$D_8 \leftarrow R_2$$

$$D_9 \leftarrow R_4$$

$$D_{10} \leftarrow R_4$$

$$D_{11} \leftarrow R_4$$

$$D_{12} \leftarrow R_4$$

$$D_{13} \leftarrow R_4$$

$$D_{15} \leftarrow R_4$$

$$D_{16} \leftarrow R_4$$

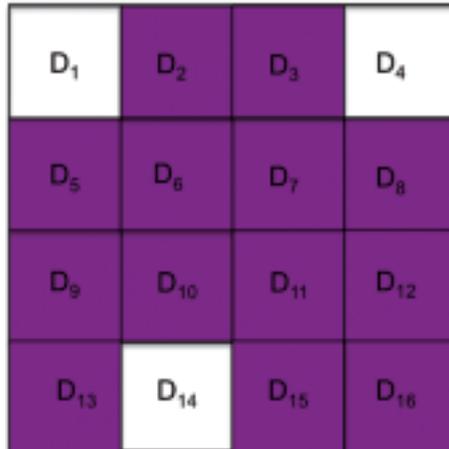
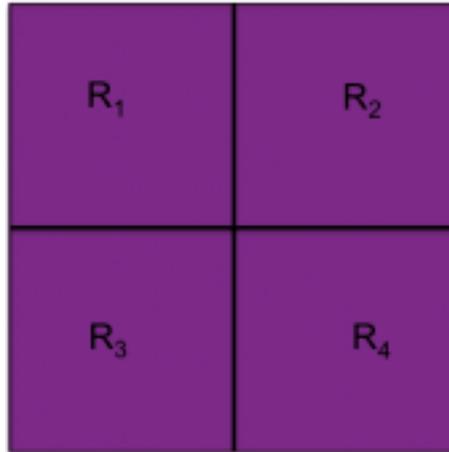
Con tali associazioni abbiamo ridotto a 13 le informazioni necessarie per la memorizzazione della figura.

3.2.2 – Processo di ricostruzione dell'immagine

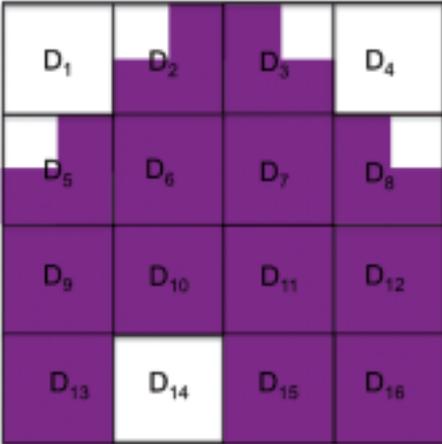
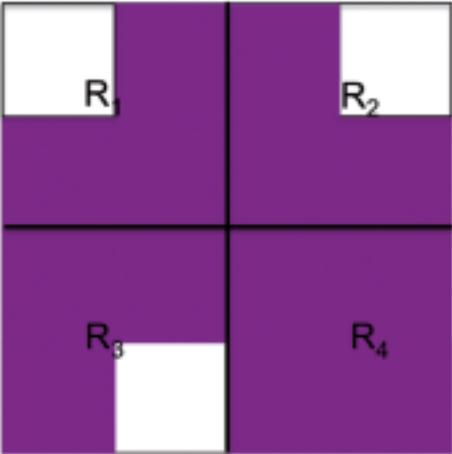
L'operatore L di ricostruzione dell'immagine è il seguente:

$$\begin{aligned} L(a) = & W_2(a \cap R_1) \cup W_3(a \cap R_2) \cup W_5(a \cap R_1) \cup W_6(a \cap R_4) \\ & \cup W_7(a \cap R_4) \cup W_8(a \cap R_2) \cup W_9(a \cap R_4) \cup W_{10}(a \cap R_4) \\ & \cup W_{11}(a \cap R_4) \cup W_{12}(a \cap R_4) \cup W_{13}(a \cap R_4) \\ & \cup W_{15}(a \cap R_4) \cup W_{16}(a \cap R_4) \end{aligned}$$

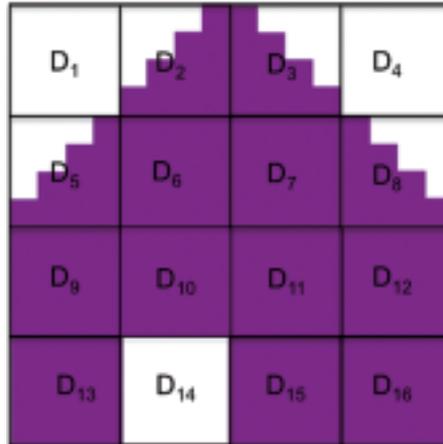
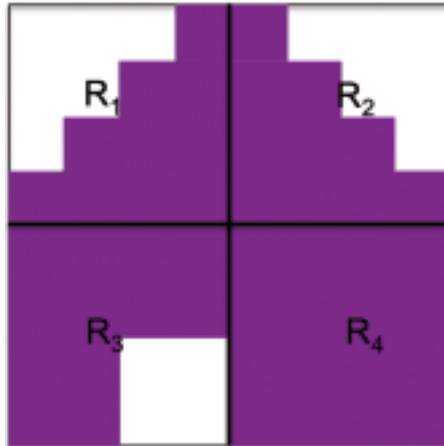
Applicando quest'operatore al quadrato pieno si ottiene:



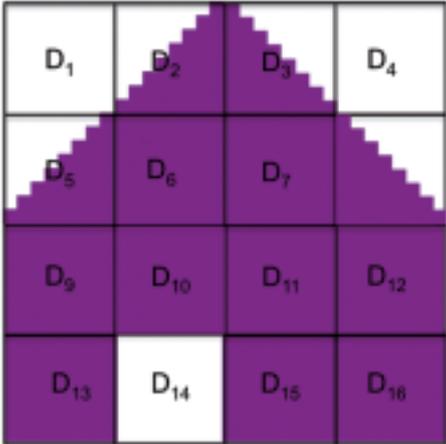
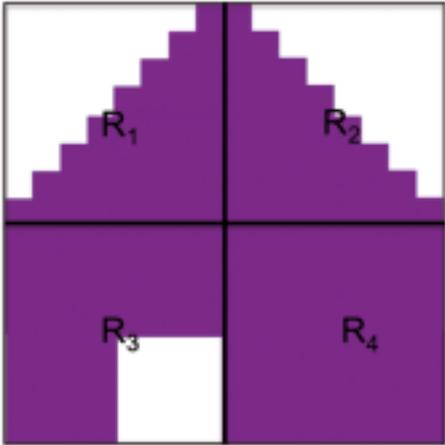
Applicando nuovamente l'operatore alla figura appena ottenuta si ha:



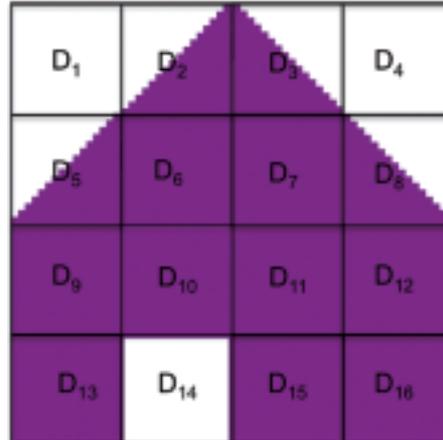
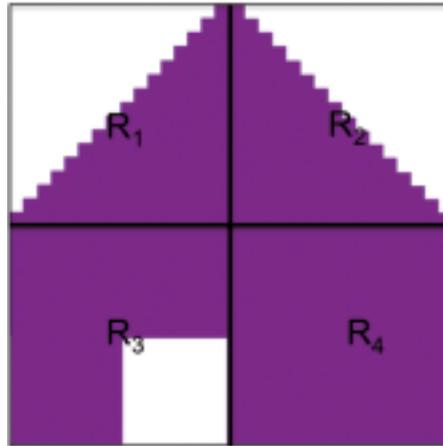
Reiterando L sulla figura ottenuta si ha:



Applicando ancora una volta L otteniamo:



Applichiamo ancora L :

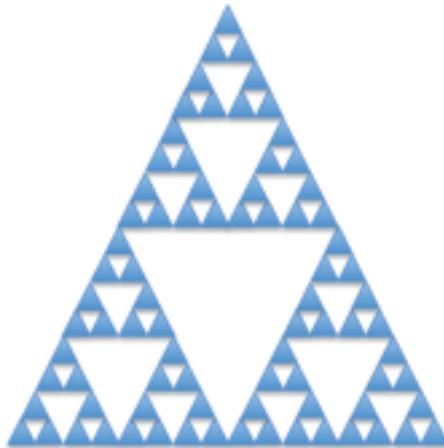


Alla fine della quinta iterazione otteniamo:



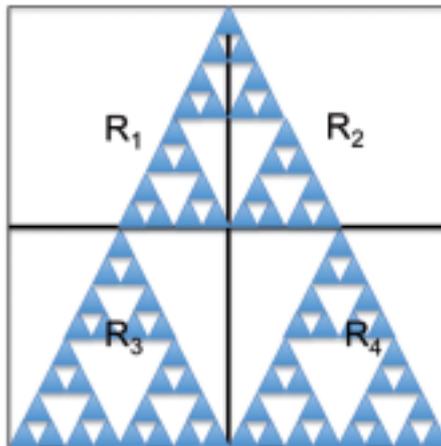
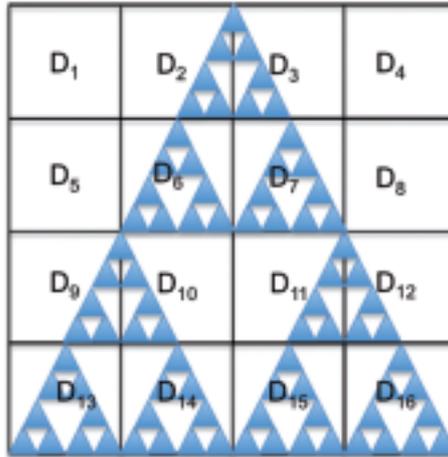
3.3 – Esempio 3

Consideriamo ora l'esempio del triangolo di Sierpinski:



3.3.1 – Processo di compressione dell'immagine

Procedendo come nell'esempio precedente, cioè costruendo le griglie 16×16 e 32×32 per il Triangolo di Sierpinski, come nelle seguenti figure



troviamo le seguenti associazioni:

$$D_2 \leftarrow R_1$$

$$D_3 \leftarrow R_2$$

$$D_6 \leftarrow R_3$$

$$D_7 \leftarrow R_3$$

$$D_9 \leftarrow R_1$$

$$D_{10} \leftarrow R_2$$

$$D_{11} \leftarrow R_1$$

$$D_{12} \leftarrow R_2$$

$$D_{13} \leftarrow R_3$$

$$D_{14} \leftarrow R_3$$

$$D_{15} \leftarrow R_3$$

$$D_{16} \leftarrow R_3$$

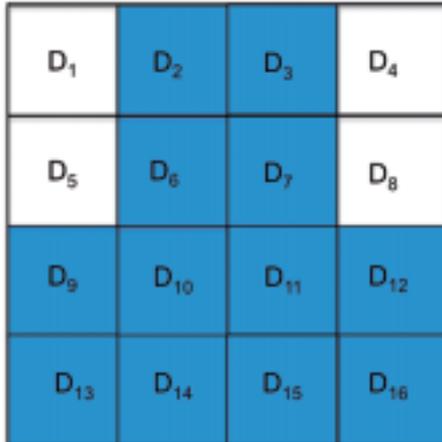
Con tali associazioni abbiamo ridotto a 12 le informazioni necessarie per la memorizzazione della figura.

3.3.2 – Processo di ricostruzione dell'immagine

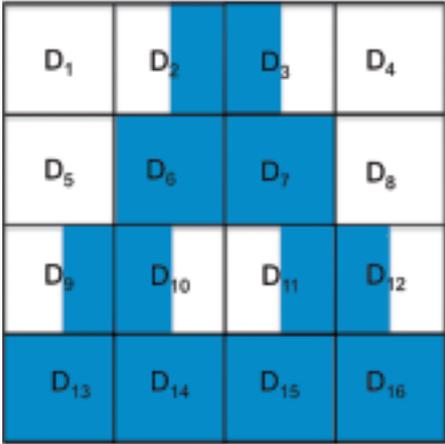
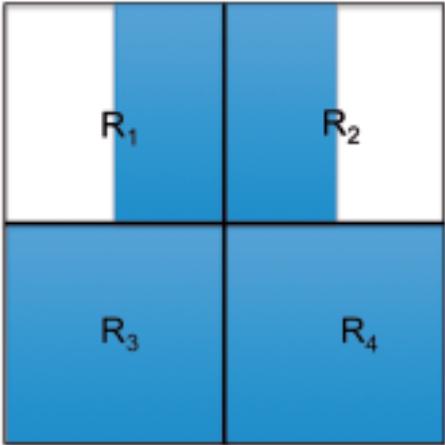
L'operatore L di ricostruzione dell'immagine è il seguente:

$$\begin{aligned} L(a) = & W_2(a \cap R_1) \cup W_3(a \cap R_2) \cup W_6(a \cap R_3) \cup W_7(a \cap R_3) \\ & \cup W_9(a \cap R_1) \cup W_{10}(a \cap R_2) \cup W_{11}(a \cap R_1) \cup W_{12}(a \cap R_2) \\ & \cup W_{13}(a \cap R_3) \cup W_{14}(a \cap R_3) \cup W_{15}(a \cap R_3) \cup W_{16}(a \cap R_3) \end{aligned}$$

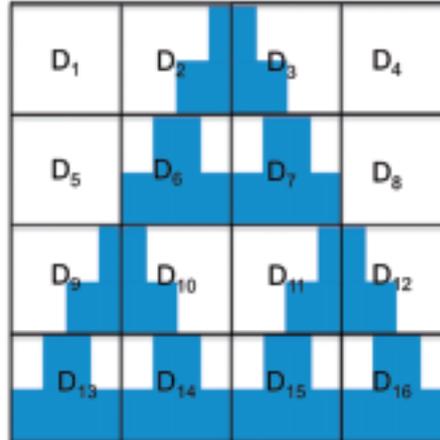
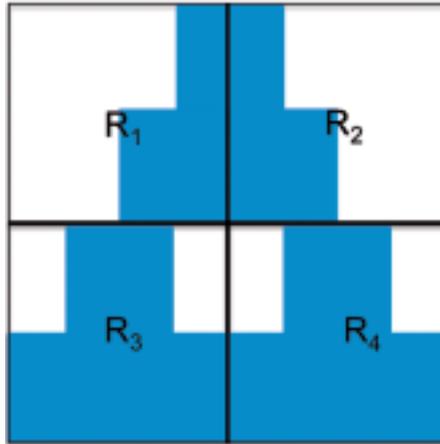
Applicando quest'operatore al quadrato pieno si ottiene:



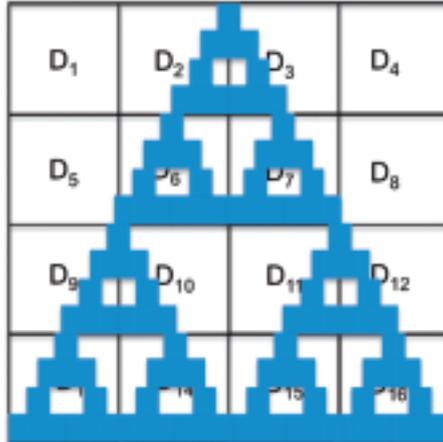
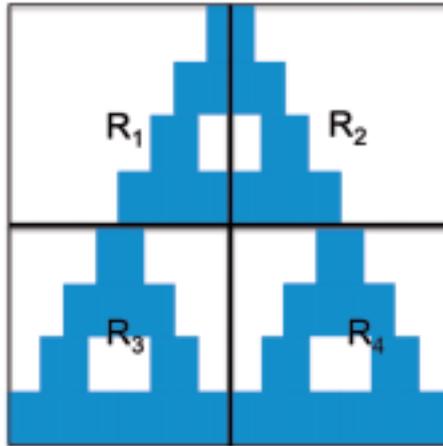
Applicando nuovamente l'operatore alla figura appena ottenuta si ha:



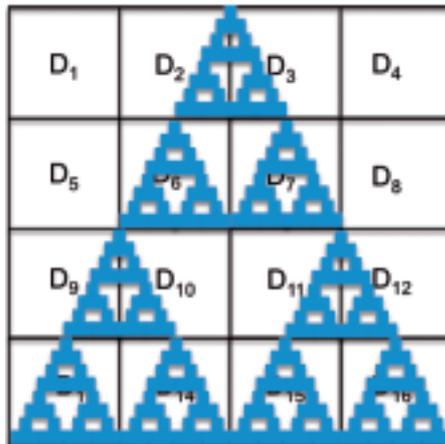
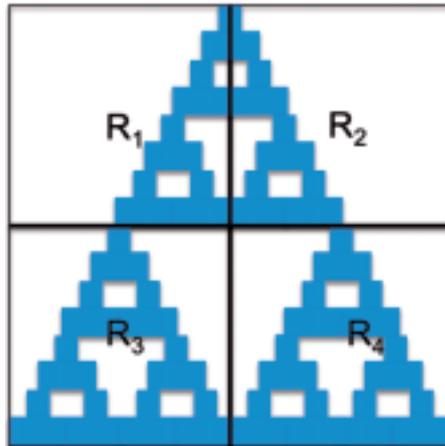
Reiterando L sulla figura ottenuta si ha:



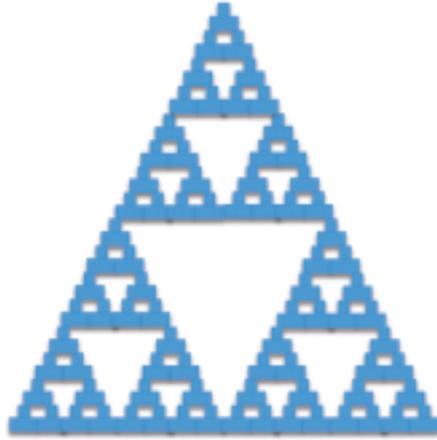
Applichiamo ancora L :



Infine, reiterando L sulla figura ottenuta si ha:



Alla fine della quinta iterazione otteniamo:



Abbiamo realizzato un video [7] per illustrare il processo di compressione e ricostruzione di un'immagine spiegato sopra.

4. – Autosimilarità debole

Abbiamo visto che la circonferenza, il cerchio e la V non sono autosimili, ovvero che data una di queste figure A , è falso che

$$A = \bigcup_{i=1}^k S_i(A)$$

con $S_i : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ similitudini contrattive per $i = 1, \dots, k$.

Ricordiamo la formula (2) che definisce una similitudine: tale relazione richiede che il quoziente

$$\frac{d(S(x), S(y))}{d(x, y)}$$

sia costante al variare di x e y .

Consideriamo ora funzioni Lipschitziane, cioè quelle in cui è richiesto che tale quoziente sia solo limitato da una costante ρ , cioè

$$\frac{d(L(x), L(y))}{d(x, y)} \leq \rho.$$

Questo vuol dire che, sotto l'azione di L , alcuni punti vengono avvicinati più di altri. Poiché ci interessano le contrazioni, vogliamo che la costante ρ sia minore di 1 e parleremo, quindi, di funzioni Lipschitziane contrattive.

Diremo che A è *debolmente autosimile* se

$$A = \bigcup_{i=1}^k L_i(A)$$

con $L_i : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ lipschitziane contrattive per $i = 1, \dots, k$.

Iniziamo col prendere in esame il caso della circonferenza. Abbiamo bisogno di una funzione lipschitziana, contrattiva, che mandi la circonferenza a coprire un pezzo della circonferenza stessa, ovvero un arco di circonferenza. Per semplificare i conti, scegliamo di lavorare con la circonferenza di centro l'origine e raggio 1:

$$C = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 = 1\}$$

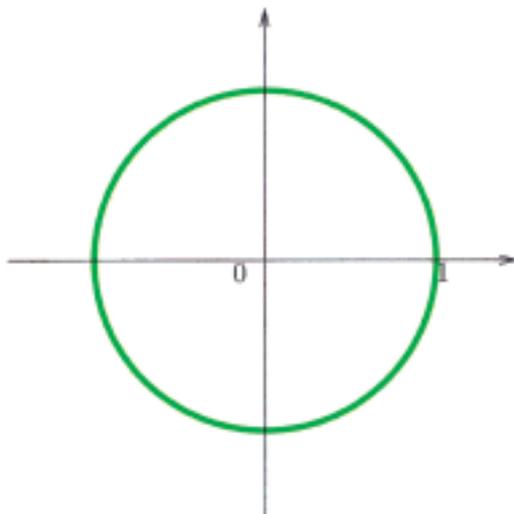


Figura 15: Circonferenza.

Consideriamo la funzione $v : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, definita nel seguente modo:

$$(4) \quad v(x, y) = \left(\frac{x}{\sqrt{2}}, \frac{\sqrt{1+y^2}}{\sqrt{2}} \right)$$

Si tratta di una funzione lipschitziana con costante di Lipschitz pari a $\frac{1}{\sqrt{2}}$. Tale funzione manda la circonferenza di partenza nell'arco di circonferenza

$$C_1 = \left\{ (x, y) \in \mathbb{R}^2 : x^2 + y^2 = 1, -\frac{1}{\sqrt{2}} \leq x \leq \frac{1}{\sqrt{2}}, y \geq 0 \right\}$$

rappresentato in figura 16:

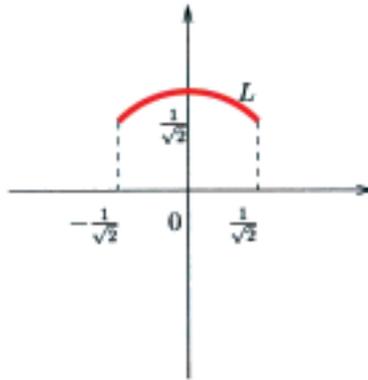


Figura 16: Arco di circonferenza.

I restanti pezzi della circonferenza saranno ottenuti mediante opportune rotazioni dell'arco appena trovato. In particolare:

$$C_2 = R_{\frac{\pi}{2}}(C_1) \quad C_3 = R_{\pi}(C_1) \quad C_4 = R_{\frac{3\pi}{2}}(C_1)$$

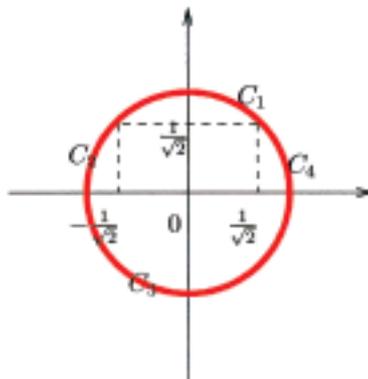


Figura 17: Circonferenza debolmente autosimile.

Ne concludiamo dunque

TEOREMA 7. – *La circonferenza è debolmente autosimile.*

Ora consideriamo il caso del cerchio. Ricordiamo che esso non è autosimile perchè il bordo del cerchio non può essere ricostruito a partire dalle sue copie rimpicciolite. Tale ostacolo può essere superato ricoprendo piccole aree vicine alla frontiera con otto lunette applicando la funzione v definita in (4) al cerchio:

$$B = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq 1\}$$

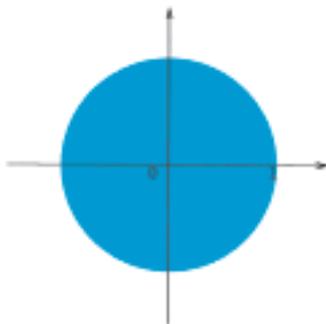


Figura 18: Cerchio.

$v(B)$ non è altro, infatti, che la seguente lunetta

$$L_1 = \left\{ (x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq 1, -\frac{1}{\sqrt{2}} \leq x \leq \frac{1}{\sqrt{2}}, y \geq \frac{1}{\sqrt{2}} \right\}$$

rappresentata in figura 19:

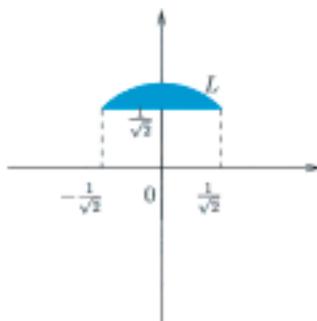


Figura 19: Lunetta.

Le restanti lunette possono essere ottenute ruotando opportunamente L_1 :

$$L_2 = R_{\frac{\pi}{2}}(L_1) \quad L_3 = R_{\pi}(L_1) \quad L_4 = R_{\frac{3\pi}{2}}(L_1)$$

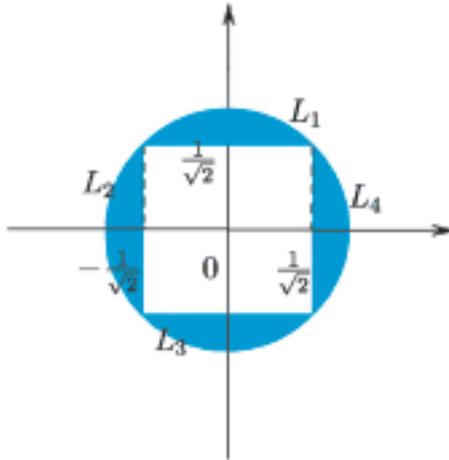


Figura 20: Lunette.

Per assicurare che nessun punto del bordo del cerchio rimanga scoperto consideriamo inoltre le lunette

$$L_5 = R_{\frac{\pi}{4}}(L_1) \quad L_6 = R_{\frac{3\pi}{4}}(L_1) \quad L_7 = R_{\frac{5\pi}{4}}(L_1) \quad L_8 = R_{\frac{7\pi}{4}}(L_1)$$

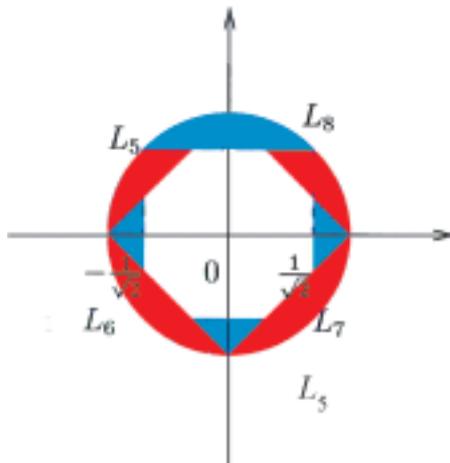


Figura 21: otto lunette.

Rimane ora da ricoprire l'ottagono regolare. Bisogna quindi trovare l'equazione della circonferenza che circoscrive l'ottagono e il cerchio corrispondente sarà $D = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq 2 - \sqrt{2}\}$ da cui si ha che

TEOREMA 8. – *Il cerchio è debolmente autosimile.*

5. – Conclusioni

Il tema trattato in questo lavoro ha costituito un valido spunto per la realizzazione di attività nell'ambito del *Progetto Lauree Scientifiche*, [6], che ha lo scopo di interessare i giovani alle scienze di base, quali matematica, fisica, chimica e scienze dei materiali, con attività extracurricolari e di orientamento all'università.

In particolare, nel marzo del 2009, ad Avezzano, è stata realizzata un'attività di sei incontri pomeridiani, da un'ora e mezza ciascuno, dal titolo *Frattali*, [4], che ha coinvolto quattro insegnanti ed una quindicina di studenti del triennio del liceo scientifico. Dopo un'introduzione del primo autore, il gruppo di lavoro si è dedicato alla codifica delle immagini con carta e matita, utilizzando la tecnica descritta nel paragrafo 3.

Successivamente, nel giugno 2013, una simile attività [5], in versione ridotta, è stata proposta da entrambi gli autori ad una ventina di studenti provenienti da licei scientifici abruzzesi, in occasione della loro visita all'Università degli Studi dell'Aquila.

Ringraziamenti. – Gli autori ringraziano il revisore anonimo per le utili osservazioni che hanno permesso di migliorare la presente esposizione. Essi sono inoltre grati al MIUR per il supporto ricevuto attraverso il Progetto Lauree Scientifiche.

RIFERIMENTI BIBLIOGRAFICI

- [1] BARNSLEY M. & HURD L., *Fractal Image Compression*, A.K. Peters, (1993).
- [2] FALCONER K.J., *The geometry of fractal sets*, Cambridge University Press, UK, (1985).
- [3] FISHER Y., *Fractal Image Compression: theory and application to digital images*, Springer Verlag, New York (1995).

- [4] http://univaq.it/~leonetti/lauree_sc/pagina1.html#frattali2009
- [5] http://univaq.it/~leonetti/lauree_sc/pagina1.html#frattali2013
- [6] <http://www.progettolaureescientifiche.eu>
- [7] https://www.youtube.com/watch?v=_HL8Qnm3XwU
- [8] HUTCHINSON J., *Fractal and Selfsimilarity*, Indiana University Mathematics Journal, 30 (1981), 713-747.
- [9] HUTCHINSON J., *Deterministic and Random Fractals in T.R.J. Bossomaier & D.G. Green (Eds) Complex Systems*, Cambridge, Cambridge University Press, (2000), 127-166.
- [10] LEONETTI F., *Fractal image compression* Mathematical Modelling (ICTMA 12): Education, Engineering and Economics; *Proceedings from the Twelfth International Conference on the Teaching of Mathematical Modelling and Applications*; Editors: C. Haines, P. Galbraith, W. Blum, S. Khan; Horwood Publishing, Chichester, UK, 2007; 377-385.
- [11] MANETTA M., *Autosimilarità e trattamento delle immagini*, Tesi di laurea in Matematica, Università degli Studi dell'Aquila, (A.A. 2005-2006).

Francesco Leonetti - Manuela Manetta

Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica

Università dell'Aquila, Via Vetoio, Coppito, I-67100 L'Aquila, Italy

leonetti@univaq.it manetta@univaq.it