

FIC: Fractal Image Compression

L. De Sano, A. Donizetti

Settembre 2015

Indice

1	Introduzione	3
2	Aspetti teorici	4
2.1	Introduzione ai frattali	4
2.2	Iterated Function Systems	5
2.3	Self-similarity nelle immagini	6
3	Aspetti Pratici ed una implementazione MATLAB	8
4	Test, benchmarks	9

1 Introduzione

Con il termine “Fractal image Compression” si va ad indicare una famiglia di tecniche di compressione di immagini (o video) basate sulle proprietà matematiche dei frattali. Tali metodi di compressione si rivelano sopra ogni altra cosa adatti a comprimere textures e immagini naturali, o, più in generale, immagini che sono caratterizzate da un elevato livello di *self-similarity* (ovvero aventi delle parti che, al netto di rotazioni e ingrandimenti/riduzioni, somigliano ad altre parti dell’immagine).

La compressione di immagini tramite frattali (così come altre, più diffuse, ad esempio *JPEG*) appartiene a quel gruppo di tecniche di compressione *lossy*, ovvero in cui la compressione dell’immagine avviene al costo di una perdita di informazione. Tuttavia, a differenza di quanto accade quando si utilizza uno dei metodi di compressione basati sui pixel (come *JPEG*, *GIF* o *MPEG*), nella compressione frattale nessuna parte dell’immagine viene effettivamente memorizzata. Ciò che viene memorizzato è invece la *struttura interna* dell’immagine (ad esempio un indice di quali parti, effettuate le dovute trasformazioni, sono simili ad altre parti).

Poiché nessun pixel dell’immagine originale viene memorizzato, la decompressione parte da un singolo pixel, di colore qualsiasi, e procede alla ricostruzione dell’immagine originale applicando iterativamente una mappa ricavata dalla struttura interna dell’immagine originale.

In questo documento tratteremo delle tecniche di compressione di immagini basate su frattali, iniziando con una panoramica teorica del loro funzionamento, proseguendo con la discussione di alcuni aspetti pratici e presentando una implementazione giocattolo realizzata in MATLAB, e concludendo infine con alcuni test che consentiranno di valutare praticità e *performances* di una libreria di compressione basata su frattali, anche in confronto con altre tecniche di compressione *lossy* maggiormente utilizzate.

2 Aspetti teorici

2.1 Introduzione ai frattali

Un frattale è un oggetto geometrico che si ripete nella sua forma, allo stesso modo, su diverse scale. Questo comportamento fa sì che ingrandendo una sua qualsiasi componente, ciò che si ottiene è una figura simile all'originale. In linea di massima, si può dire che perché l'insieme F sia considerato un frattale, F dovrebbe avere almeno le seguenti proprietà:

- F ha dettagli ad ogni scala d'ingrandimento;
- F gode di autosimilarità (a qualunque scala si osservi, presenta sempre le stesse caratteristiche globali);
- la dimensione frattale di F è maggiore della sua dimensione topologica;
- esiste un algoritmo relativamente semplice per costruire F .

TODO: definire dimensione frattale e topologica

<http://www.vanderbilt.edu/AnS/psychology/cogsci/chaos/workshop/Fractals.html>

Un esempio di oggetto geometrico secondo i principi di costruzione di un frattale è che rispetta le proprietà elencate è la curva di *Koch*. La costruzione comincia con una linea di lunghezza 1 chiamata *initiator*. Da questa linea si rimuove il terzo centrale e lo si sostituisce con due linee della stessa lunghezza della parte rimossa. Questa nuova forma viene chiamata *generator*. La prima parte della costruzione è mostrata in figura 2.1.

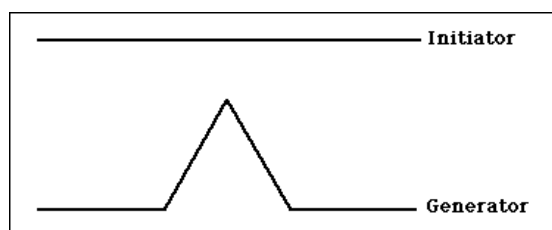


Figura 2.1: Initiator e generator per la curva di Koch

La regola può essere nuovamente applicata su ogni linea, così da andare a sostituirla ogni volta come fanno nel passaggio da *initiator* a *generator*. Il secondo livello è visibile in figura 2.2.

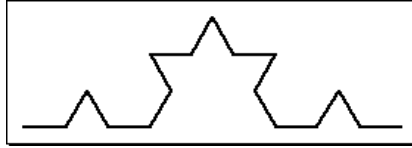


Figura 2.2: Livello 2 per la curva di Koch

Una volta che la procedura è avviata può proseguire a piacimento. Il terzo e il quarto livello sono visibili nelle figure 2.3 e 2.4 rispettivamente.



Figura 2.3: Livello 3 per la curva di Koch

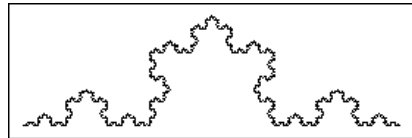


Figura 2.4: Livello 4 per la curva di Koch

2.2 Iterated Function Systems

Uno strumento matematico che si rivelerà fondamentale nella fase di decompressione delle immagini trattate con un metodo di compressione frattale è quello degli *Iterated Function Systems*, che andiamo per questo motivo a descrivere qui.

Senza eccedere in formalità, definiamo un *Iterated Function System* (IFS) come una collezione di trasformazioni contrattive $\{w_i : \mathbb{R}^2 \rightarrow \mathbb{R}^2\}_{i=1,\dots,n}$ che mappa il piano su se stesso. Questa collezione di trasformazioni definisce una mappa

$$W(\cdot) = \bigcup_{i=1}^n w_i(\cdot)$$

Tale mappa è applicata ad insiemi di punti, ed il risultato è definito in questa maniera: dato un insieme di punti $S \in \mathbb{R}^2$, calcoliamo $w_i(S)$ per ogni i (ovvero produciamo i copie ridotte di S), e poi calcoliamo l'unione dei risultati (ovvero, mettiamo insieme tutte le copie ridotte), ottenendo così un nuovo insieme $S' = W(S)$.

W così definito è una mappa tra sottoinsiemi di \mathbb{R}^2 (che d'ora in avanti per noi saranno “immagini”), e gode di due importanti (e rilevanti per la compressione frattale) proprietà, che andiamo ad enunciare senza dimostrazione.

Proprietà 1 *Se tutte le w_i sono contrattive, allora W è contrattiva in uno spazio di sottoinsiemi del piano.*

Proprietà 2 *Fissata una mappa W , esiste una immagine x_W , chiamata attrattore per W , tale che*

- $W(x_W) = x_W$
- data una qualunque immagine di partenza S_0 , vale che

$$x_W \equiv \lim_{n \rightarrow \infty} W^n(S_0)$$

ovvero l'applicazione ripetuta per n volte di W porta ad ottenere x_W , per n sufficientemente grande, ed indipendentemente dalla scelta dell'immagine di partenza.

- x_W è unica. Se una qualsiasi immagine S soddisfa $W(S) = S$, allora S è l'attrattore per W .

La seconda di queste proprietà è nota come *Contractive Mapping Fixed-Point Theorem*.

2.3 Self-similarity nelle immagini

Immagini come oggetti matematici

Per poter applicare la teoria degli *IFS* alle immagini come comunemente le intendiamo (ovvero, in sostanza, matrici di pixels rappresentati da 1 byte – *greyscale* – oppure più di uno – es. RGB –, è necessario formalizzare in qualche modo il concetto di “immagine digitale”, cercando di inserirlo in un contesto matematico che ci consenta di manipolarla utilizzando di strumenti messi a disposizione dall'algebra. Per semplicità, ci limiteremo qui a considerare le immagini in *grayscale* (un singolo canale di colore, l'intensità indicata con un numero da 0 a 255).

Consideriamo un'immagine in scala di grigi, memorizzata come matrice di pixel, 1 byte per ogni pixel. Non è difficile immaginare di poter rappresentare tale immagine utilizzando una funzione $f : \mathbb{R}^2 \rightarrow \{1, 2, \dots, 255\}$, in modo tale che ad ogni pixel alle coordinate (x, y) sia fatto corrispondere un punto (x, y) nel piano, al quale a sua volta viene associata un'altezza z , che corrisponde al valore di grigio del pixel in questione.

Ad esempio, ad un'immagine con il primo pixel in basso a sinistra avente livello di grigio 177, viene fatta corrispondere una funzione f che, per quanto riguarda quel preciso punto, avrà $f(0, 0) = 177$. Effettuando questa operazione su tutti i pixel dell'immagine originale, è possibile definire completamente f su tutta la parte del piano di nostro interesse (e, in genere, si riduce per convenzione l'immagine ad avere dominio $[0, 1] \times$

$[0, 1]$). Nella figura qui sotto riportata si può vedere un esempio del risultato finale della procedura, applicata ad una immagine grayscale 128×128 pixels.

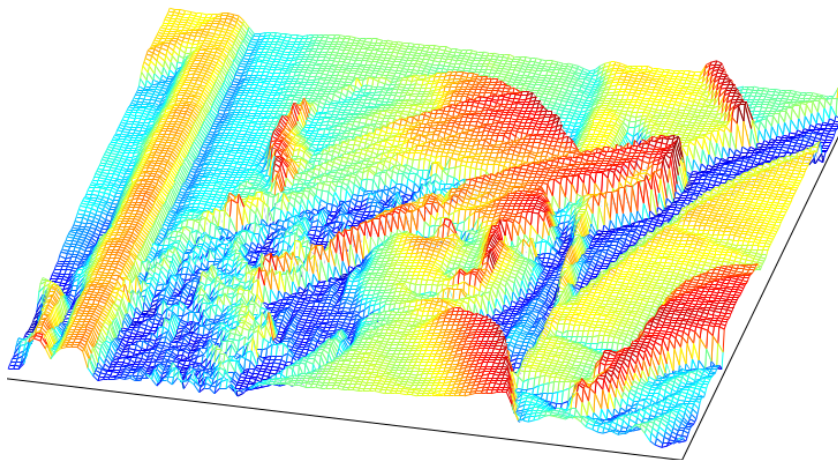


Figura 2.5: lena grayscale, 128×128 pixels, trasformata in funzione $[0, 1]^2 \rightarrow \{0, \dots, 255\}$

La trasformazione descritta ci permette di ignorare l’aspetto “visivo” della modalità in cui una immagine è memorizzata, e ci consente di identificare una immagine con la funzione f associata, su cui si andrà ad operare durante il procedimento di compressione.

Similarità tra due immagini

Un secondo concetto che è necessario introdurre per poter operare sulle immagini durante il procedimento di compressione e decompressione è quello di “distanza” tra due immagini. La contrattività di una mappa W (ovvero, intuitivamente, il fatto che essa rende le immagini più “piccole”) può essere verificata solo se si ha a disposizione una qualche metrica sulla base della quale è possibile valutare la distanza tra due sottoinsiemi del piano (ovvero, nel nostro caso, due immagini).

Date due immagini f, g , definiamo la loro distanza $d(f, g)$, utilizzando la media quadratica, come

$$d(f, s) = \sqrt{\int_{[0,1]^2} f(x, y) - g(x, y) \, dx dy}$$

La metrica ci fornisce una maniera di valutare la distanza tra due immagini pesando in maniera uniforme tutti i punti in esse contenuti.

3 Aspetti Pratici ed una implementazione MATLAB

4 Test, benchmarks