http://math.unipa.it/cvalenti/htm/tesi.htm

Analisi dati, algoritmi paralleli e evolutivi (anche in ambiente Android) per informatica forense; tomografia; diagnostica medica; elaborazione, interpretazione e compressione di immagini

Un algoritmo evolutivo per il sottocampionamento sparso. G.Albanese. 2007/08
Un algoritmo genetico parallelo per la tomografia discreta tridimensionale. G.Marfia. 2010/11
Preprocessamento genetico per la compressione di immagini. N.Messina. 2010/11
Simmetria radiale e regioni di interesse. P.Cataldo. 2010/11
Individuazione di immagini in database pittorici. A.Daidone. 2010/11
Analisi di immagini digitali in capillaroscopia. S.Mangiaracina. 2011/12
Informatica forense. R.Corso. 2011/12
Informatica forense: stima dei coefficienti di quantizzazione JPEG. G.Di Martino. 2011/12
Un sistema per la video-scrittura per soggetti locked-in. V.Dispensa. 2012/13
Filtro cartoon per immagini e video. D.Guastella. 2012/13
Informatica forense: Creazione e analisi del pattern noise di una fotocamera digitale. A.Ianni'. 2012/13
Simple view conversation. Un ambiente di videofonia mobile per soggetti audiolesi. A.Mineo. 2012/13
Un approccio evolutivo per la colorazione dei grafi. S.Caruso. 2012/13
Algoritmi per la compressione dei segnali audio. F.Castelli. 2012/13
Algoritmi di riduzione del rumore di segnali mono e bidimensionali. S.Pantina. 2013/14
Un’interfaccia grafica per la realizzazione di filmati con simulazione neuron. G.Di Dio. 2013/14
Un algoritmo genetico per l'individuazione dei punti di Steiner. M.Lo Coco. 2013/14
Un algoritmo genetico per la risoluzione del problema della sfera di Thomson. C.Gervasi. 2013/14
Sottocampionamento sparso per la compressione di segnali audio. D.Scardina. 2013/14
Crawling e indicizzazione delle immagini digitali sul web. A.Tocco. 2013/14
Individuazione dei volti e riconoscimento per fascia d'età. V.Cimino. 2014/15
Tecniche veloci di Phong shading e illuminazione. F.Guastella. 2014/15
Tecniche di Image Inpainting. M.Egorenkova. 2014/15
Un approccio evolutivo al proactive means-end reasoning. F.Li Manni. 2014/15
Individuazione di volti e mimetizzazione. G.Luciano. 2014/15
Visualizzazione scientifica di dati provenienti da simulazioni realistiche di neuroni: il caso di una rete di cellule cardiache. A.Ienna. 2015/16
Multiframe super-resolution. P.Colaianni. 2015/16
Rilevamento di artefatti tampering in immagini digitali. N.D'Agostino. 2015/16
Riconoscimento dei gesti per interfacce uomo-macchina. G.Liggio. 2015/16

Compressione

Gli algoritmi di compressione per immagini digitali devono tener conto non solamente del fattore di compressione, ma anche dell'errore eventualmente introdotto.

Le misure di compressione normalmente usate misurano:

<u>bit per pixel</u>: $bpp = \frac{C}{N}$

<u>rapporto di compressione</u>: $ratio = \frac{kN}{C}$

dove *C* indica la dimensione in *bit* del file compresso, *N* indica il numero di pixel e *k* è il numero di bit per pixel nell'immagine originale. Di solito le informazioni aggiuntive (*header*) non sono incluse nei calcoli.

Misure di qualità

L'errore (distorsione) introdotto dagli algoritmi di compressione (*lossy*) è quantitativamente misurabile tramite:

mean absolute error: $MAE = \frac{1}{N}\sum_{i=1}^{N}\left|g_i - g'_i\right|$

mean square error: $MSE = \frac{1}{N}\sum_{i=1}^{N}\left(g_i - g'_i\right)^2$

peak-to-peak signal to noise ratio: $PSNR = 10\log_{10}\left[\frac{(G-1)^2}{MSE}\right]$

Purtroppo queste misure non coincidono con la normale valutazione personale. Ad esempio, l'occhio umano non confronta i singoli pixel, ma permette una stima qualitativa della distorsione globale. Inoltre, queste misure non confrontano artefatti come i blocchi o la sfocatura.

Compressione


Input (bpp = 8.00; *MSE* = 0.00)


JPG (bpp = 1.00; *MSE* = 17.26)


JPG (bpp = 0.50; *MSE* = 33.08)


JPG (bpp = 0.25; *MSE* = 79.11)

Misure di qualità

Confrontiamo separatamente la luminosità, il contrasto e le strutture di due immagini, normalizzate con funzione *z-score*:

$$z(I) = \frac{I - \mu}{\sigma}$$

$$d_\ell = \frac{2\mu_1\mu_2 - \ell}{\mu_1^2 + \mu_2^2 + \ell} \qquad d_c = \frac{2\sigma_1\sigma_2 - c}{\sigma_1^2 + \sigma_2^2 + c} \qquad d_s = \frac{\sigma_{12} - s}{\sigma_1\sigma_2 + s}$$

dove μ_1, μ_2, σ_1 e σ_2 sono, rispettivamente, le medie e le deviazioni standard delle immagini e le quantità ℓ, c e s sono sufficientemente piccole da impedire instabilità quando $\mu_1^2+\mu_2^2=0$, $\sigma_1^2+\sigma_2^2=0$ e $\sigma_1\sigma_2=0$.

Infine, indichiamo con σ_{12} la misura di correlazione:

$$\sigma_{12} = \frac{1}{N}\sum_{i=1}^{N}(g_i - \mu_1)(g_i' - \mu_2)$$

Misure di qualità

Confrontiamo separatamente la luminosità, il contrasto e le strutture di due immagini, normalizzate con funzione *z-score*:

$$z(I) = \frac{I - \mu}{\sigma}$$

$$\boxed{d_\ell = \frac{2\mu_1\mu_2 - \ell}{\mu_1^2 + \mu_2^2 + \ell}} \qquad d_c = \frac{2\sigma_1\sigma_2 - c}{\sigma_1^2 + \sigma_2^2 + c} \qquad d_s = \frac{\sigma_{12} - s}{\sigma_1\sigma_2 + s}$$

dove μ_1, μ_2, σ_1 e σ_2 sono, rispettivamente, le medie e le deviazioni standard delle immagini e le quantità ℓ, c e s sono sufficientemente piccole da impedire instabilità quando $\mu_1^2+\mu_2^2=0$, $\sigma_1^2+\sigma_2^2=0$ e $\sigma_1\sigma_2=0$.

Infine, indichiamo con σ_{12} la misura di correlazione:

$$\sigma_{12} = \frac{1}{N}\sum_{i=1}^{N}\left(g_i - \mu_1\right)\left(g_i' - \mu_2\right)$$

Analisi di Immagini Digitali - A.A. 2016–2017 - cesare.valenti@unipa.it

Misure di qualità

Confrontiamo separatamente la luminosità, il contrasto e le strutture di due immagini, normalizzate con funzione *z-score*:

$$z(I) = \frac{I - \mu}{\sigma}$$

$$d_\ell = \frac{2\mu_1\mu_2 - \ell}{\mu_1^2 + \mu_2^2 + \ell} \qquad \boxed{d_c = \frac{2\sigma_1\sigma_2 - c}{\sigma_1^2 + \sigma_2^2 + c}} \qquad d_s = \frac{\sigma_{12} - s}{\sigma_1\sigma_2 + s}$$

dove μ_1, μ_2, σ_1 e σ_2 sono, rispettivamente, le medie e le deviazioni standard delle immagini e le quantità ℓ, c e s sono sufficientemente piccole da impedire instabilità quando $\mu_1^2+\mu_2^2=0$, $\sigma_1^2+\sigma_2^2=0$ e $\sigma_1\sigma_2=0$.

Infine, indichiamo con σ_{12} la misura di correlazione:

$$\sigma_{12} = \frac{1}{N}\sum_{i=1}^{N}(g_i - \mu_1)(g'_i - \mu_2)$$

Misure di qualità

Confrontiamo separatamente la luminosità, il contrasto e le strutture di due immagini, normalizzate con funzione *z-score*:

$$z(I)=\frac{I-\mu}{\sigma}$$

$$d_\ell=\frac{2\mu_1\mu_2-\ell}{\mu_1^2+\mu_2^2+\ell} \qquad d_c=\frac{2\sigma_1\sigma_2-c}{\sigma_1^2+\sigma_2^2+c} \qquad \boxed{d_s=\frac{\sigma_{12}-s}{\sigma_1\sigma_2+s}}$$

dove μ_1, μ_2, σ_1 e σ_2 sono, rispettivamente, le medie e le deviazioni standard delle immagini e le quantità ℓ, c e s sono sufficientemente piccole da impedire instabilità quando $\mu_1^2+\mu_2^2=0$, $\sigma_1^2+\sigma_2^2=0$ e $\sigma_1\sigma_2=0$.

Infine, indichiamo con σ_{12} la misura di correlazione:

$$\sigma_{12}=\frac{1}{N}\sum_{i=1}^{N}\left(g_i-\mu_1\right)\left(g_i'-\mu_2\right)$$

Misure di qualità

La misura di somiglianza strutturale è definita come:

$$SSIM = d_{\ell}^{\alpha} \times d_{c}^{\beta} \times d_{s}^{\gamma}$$

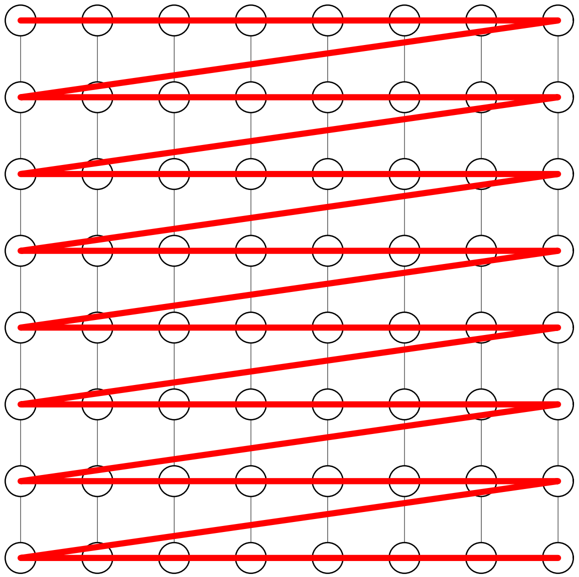
Di solito, si impongono $\alpha=\beta=\gamma=1$ e $s=c/2$, ottenendo:

$$SSIM = \frac{\left(2\mu_1\mu_2 - \ell\right)\left(2\sigma_{12} + c\right)}{\left(\mu_1^2 + \mu_2^2 + \ell\right)\left(\sigma_1^2 + \sigma_2^2 + c\right)}$$

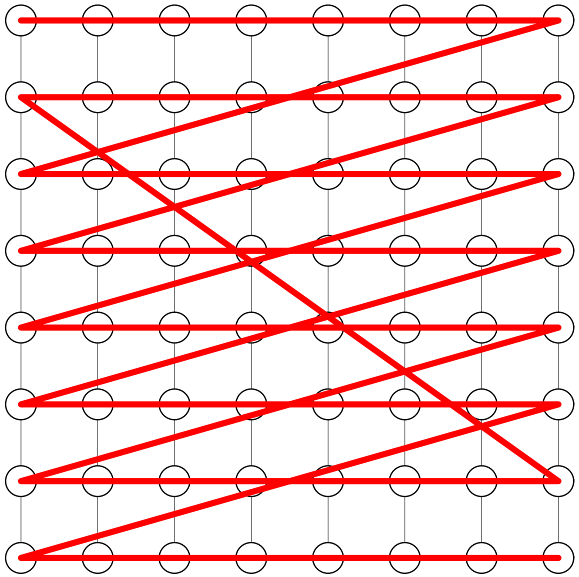
Questa misura non considera i singoli pixel, ma confronta le strutture principali.

Una migliore risposta si ottiene applicando *SSIM* a livello locale, piuttosto che globale.

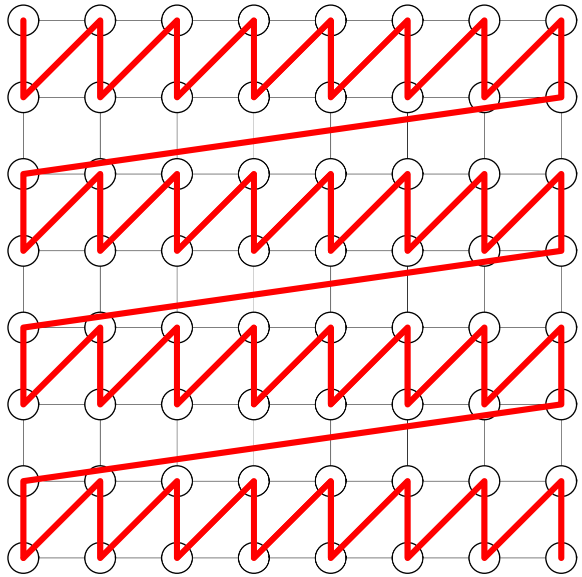
Zigzag-ing



raster

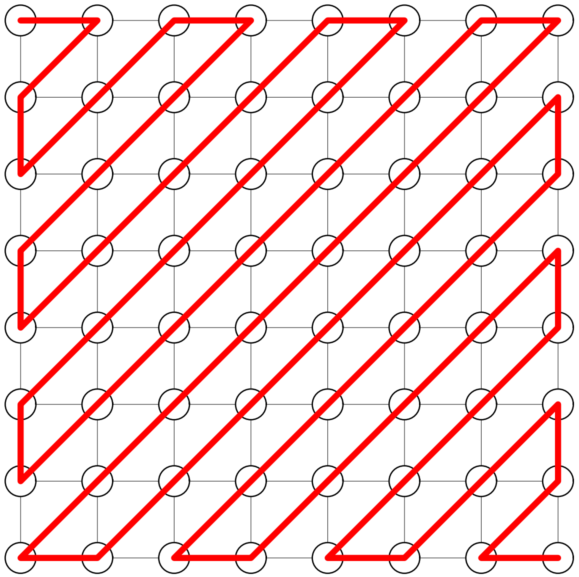


interlaced

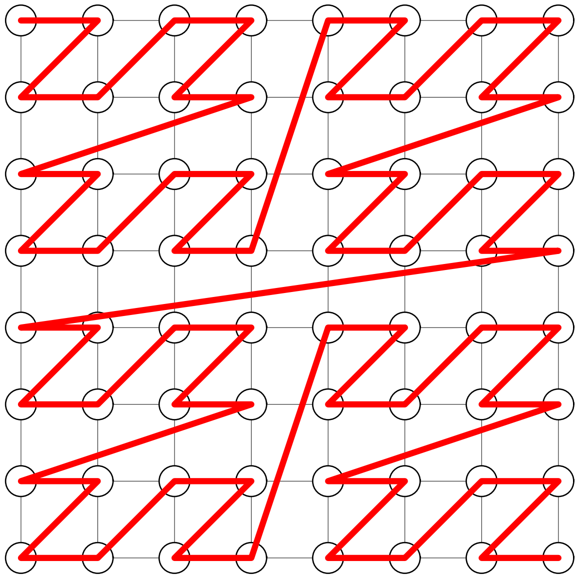


mixed

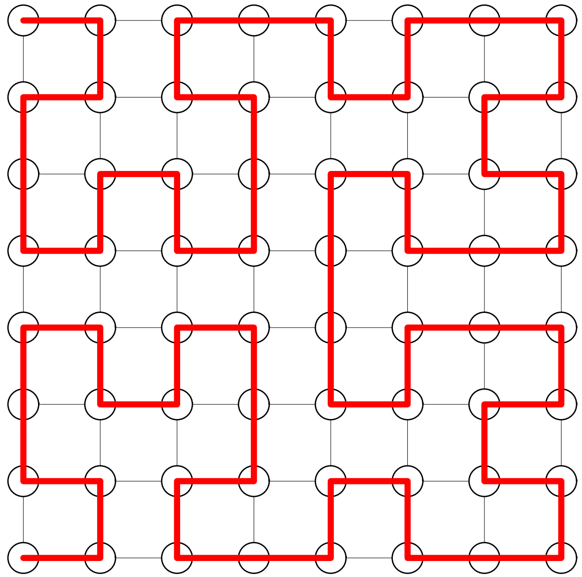
Zigzag-ing



jpg-like

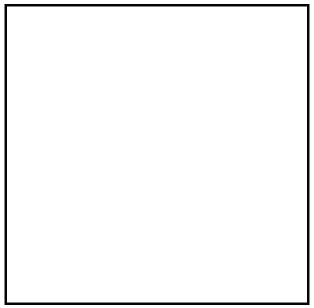


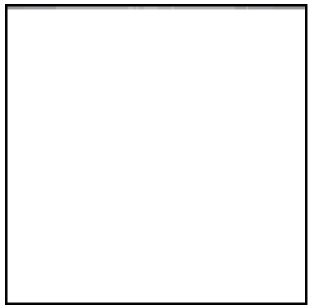
Morton

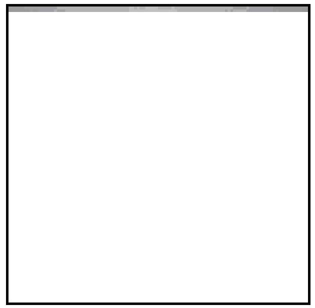


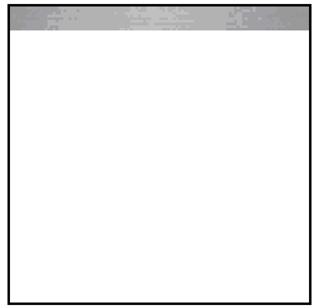
Peano-Hilbert

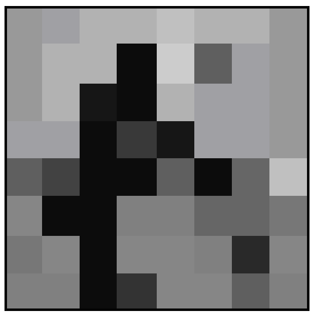
Codifica progressiva / a blocchi

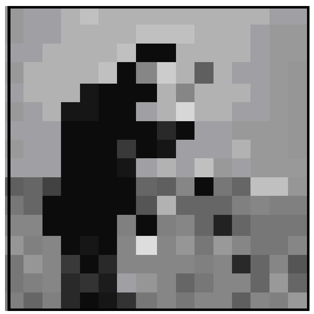

0.1%


0.5%


2.1%


8.3%


0.1%


0.5%


2.1%


8.3%

Segmentazione

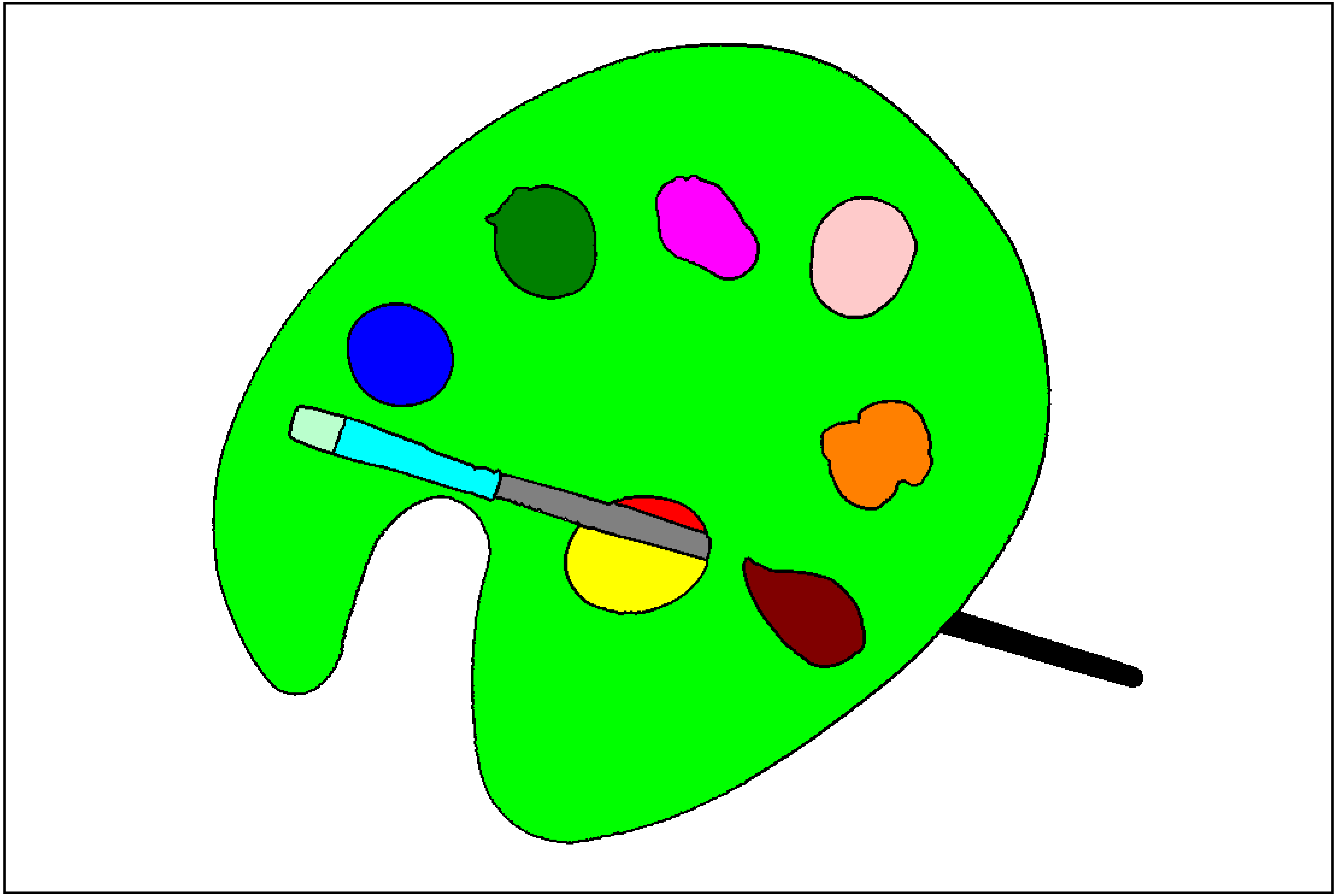
La segmentazione automatica è uno dei compiti più ardui dell'analisi di immagini digitali, poiché spesso presuppone la comprensione dell'immagine stessa. Formalmente la segmentazione di immagini è un partizionamento dell'immagine I in regioni $R_1, \ldots, R_n$ tali che:

- le regioni sono uniformi: $P(R_i)$=True
- le regioni <u>confinanti</u> hanno proprietà differenti: $P(R_i) \neq P(R_j)$, con $i \neq j$
- il ricoprimento è completo: $\bigcup_i R_i = I$
- le regioni sono a due a due disgiunte: $R_i \bigcap_{i \neq j} R_j = \varnothing$

Segmentazione

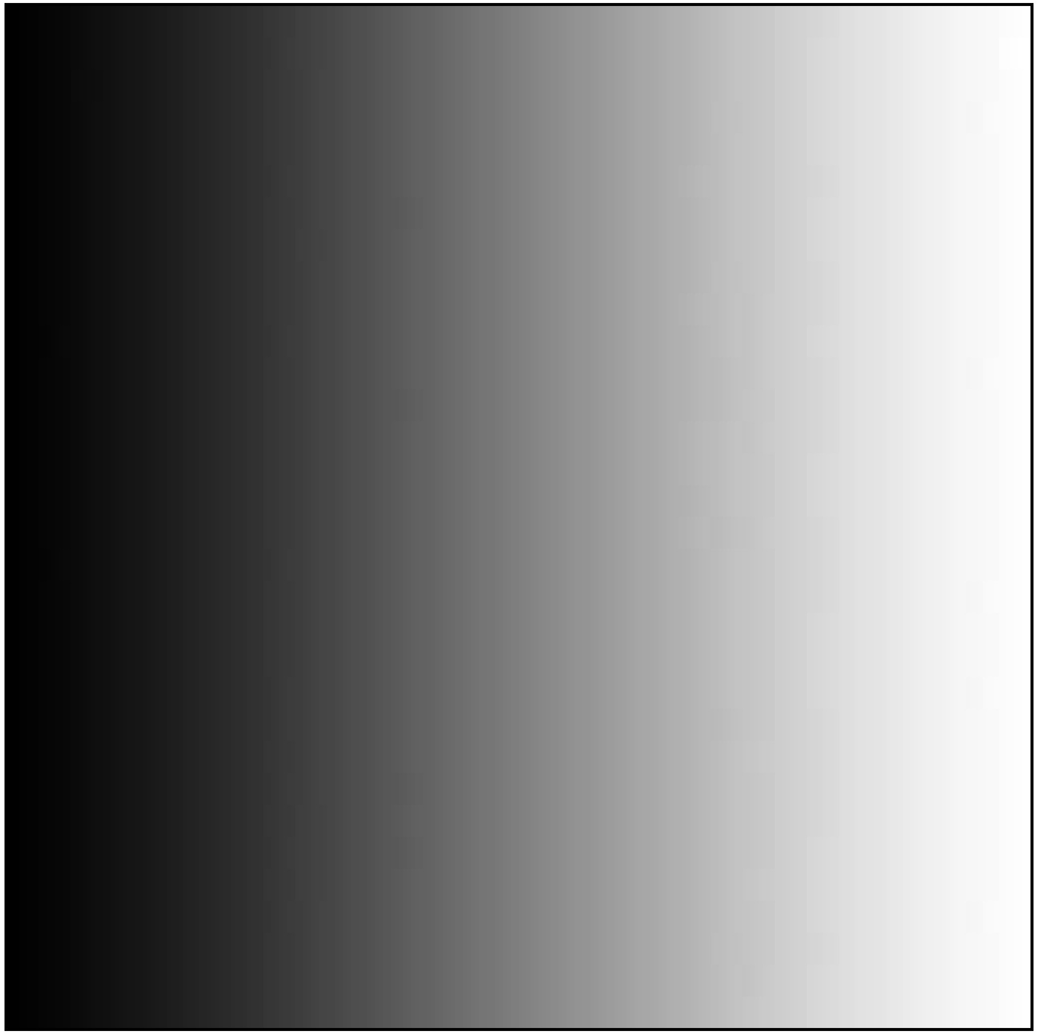
Quante componenti sono presenti in questa immagine?

Segmentazione



Quante componenti sono presenti in questa immagine?

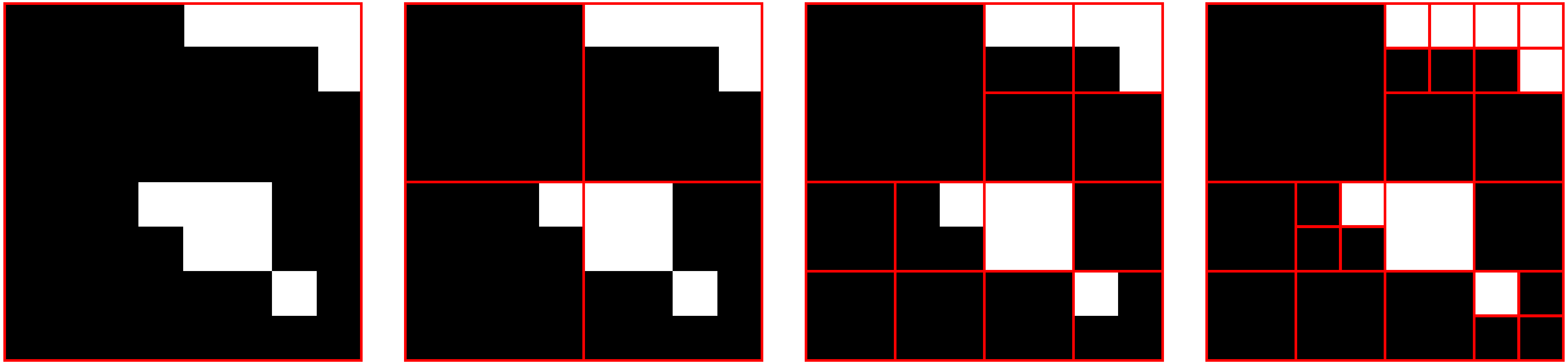
Segmentazione



Quante componenti sono presenti in questa immagine?

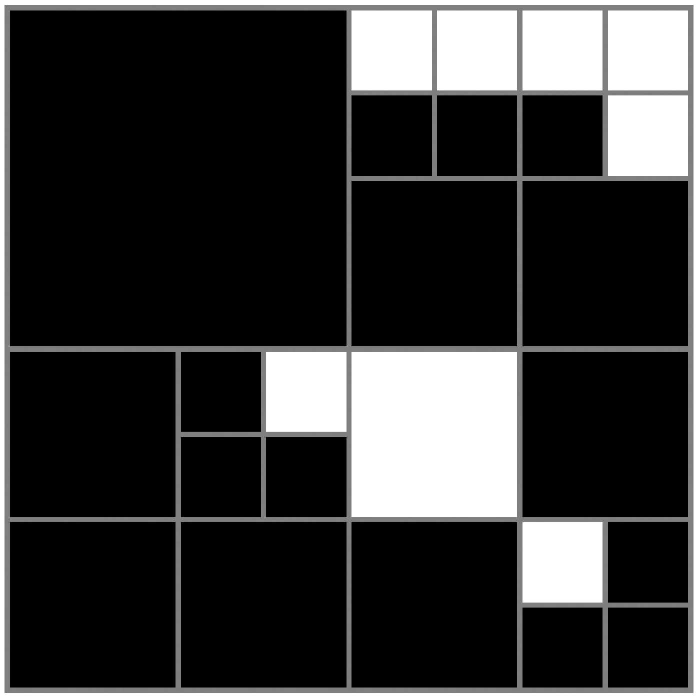
Quadtree

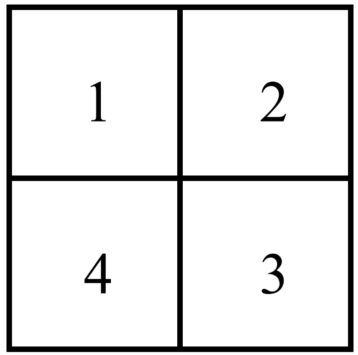
I quadtree sono strutture piramidali in cui l'immagine è divisa in quattro quadranti ad ogni livello, fino al raggiungimento di zone uniformi.

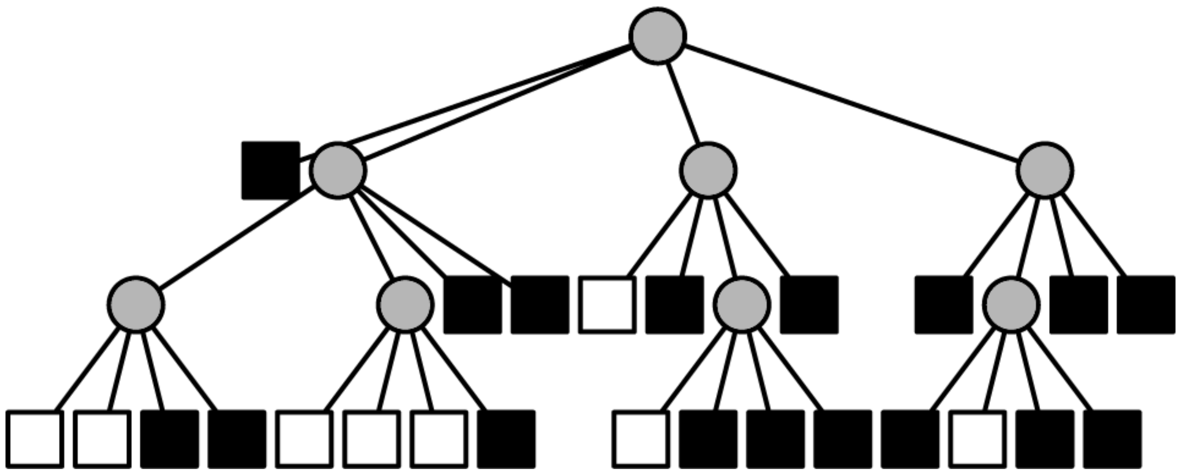


Quadtree

I quadtree sono strutture piramidali in cui l'immagine è divisa in quattro quadranti ad ogni livello, fino al raggiungimento di zone uniformi. Esiste una corrispondenza tra quadtree e i 4-alberi.

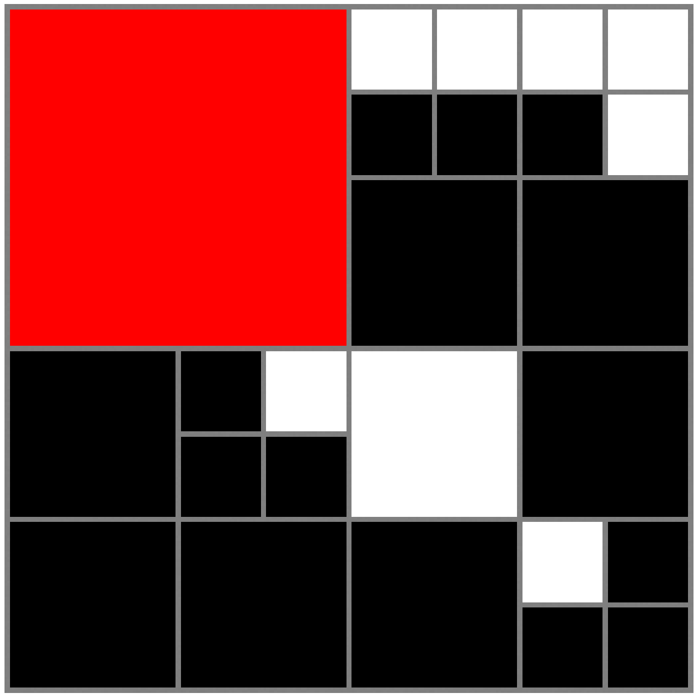


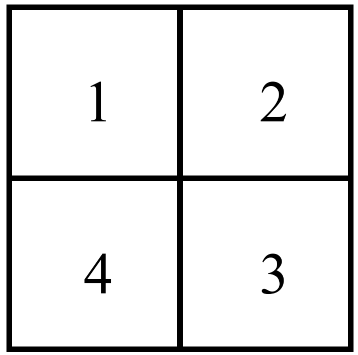


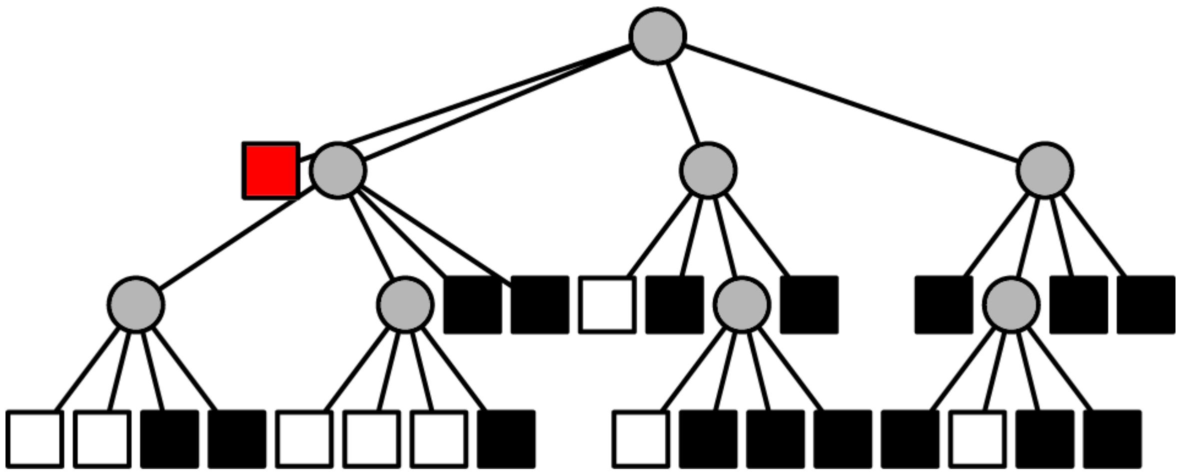


Quadtree

I quadtree sono strutture piramidali in cui l’immagine è divisa in quattro quadranti ad ogni livello, fino al raggiungimento di zone uniformi. Esiste una corrispondenza tra quadtree e i 4-alberi.

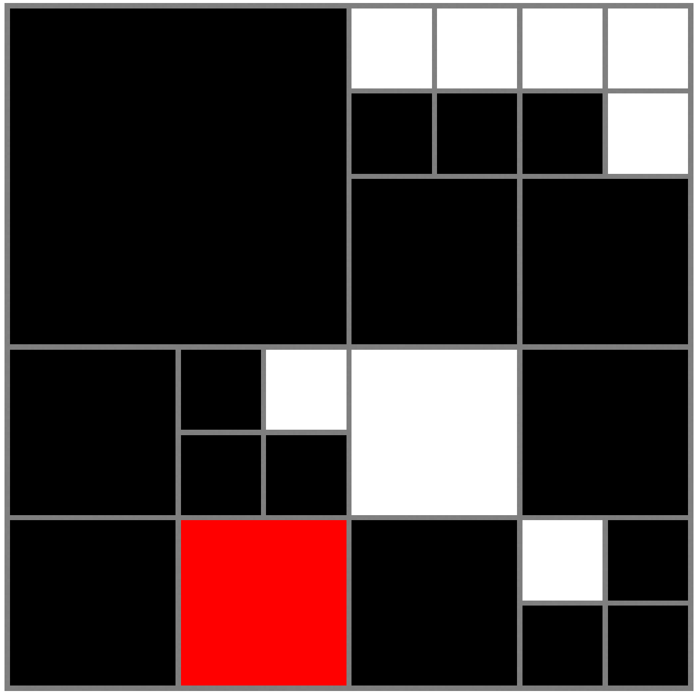


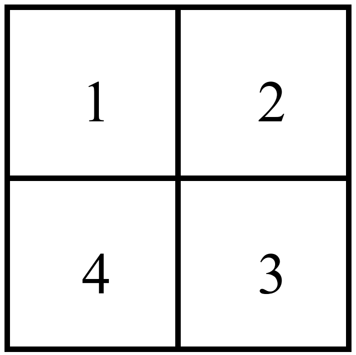


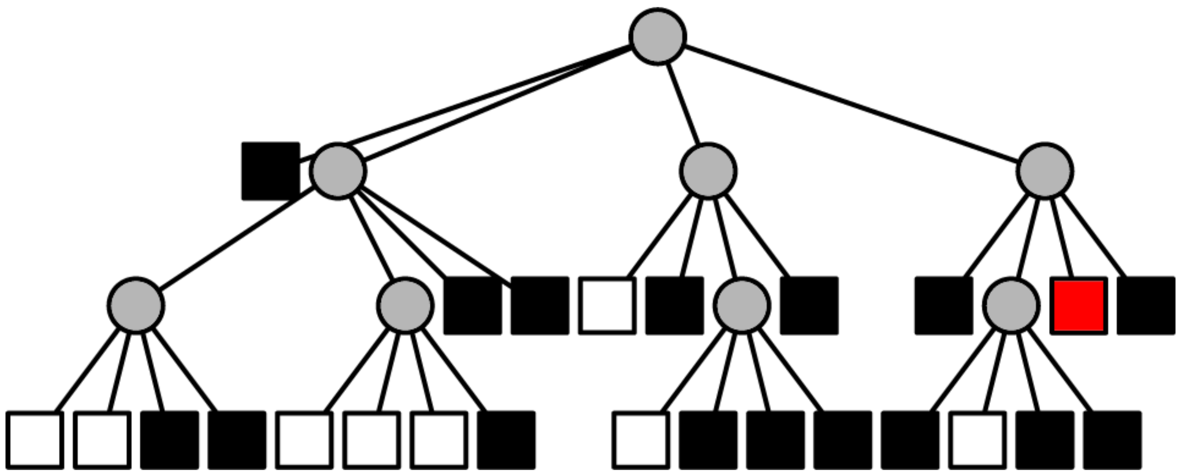


Quadtree

I quadtree sono strutture piramidali in cui l'immagine è divisa in quattro quadranti ad ogni livello, fino al raggiungimento di zone uniformi. Esiste una corrispondenza tra quadtree e i 4-alberi.

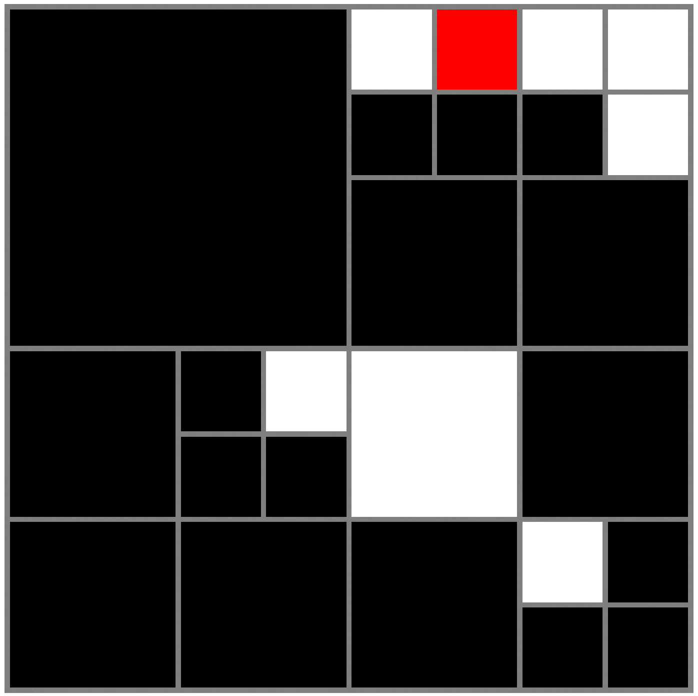


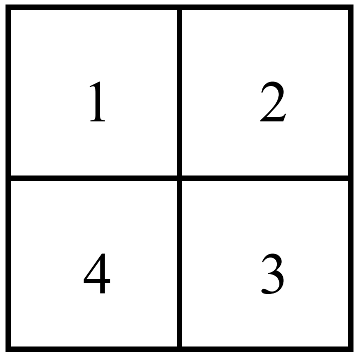


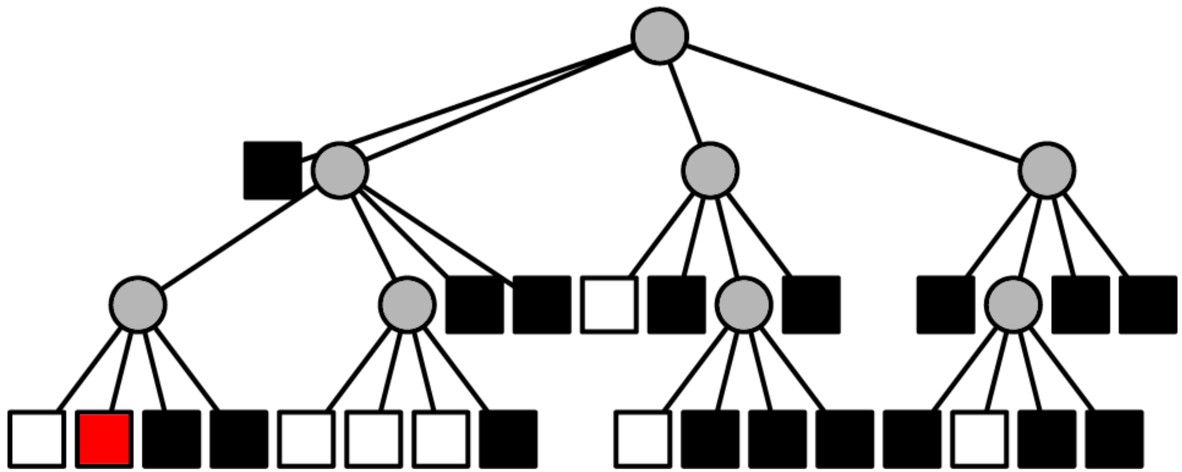


Quadtree

I quadtree sono strutture piramidali in cui l'immagine è divisa in quattro quadranti ad ogni livello, fino al raggiungimento di zone uniformi. Esiste una corrispondenza tra quadtree e i 4-alberi.

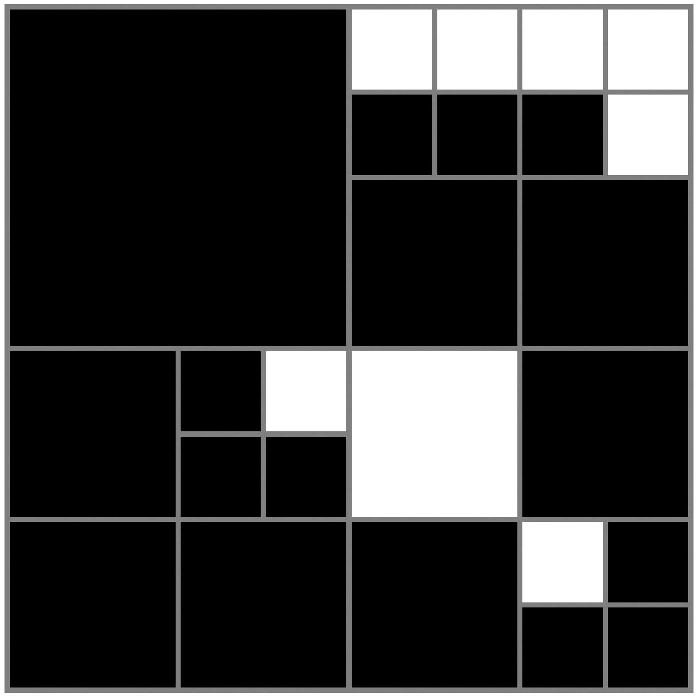


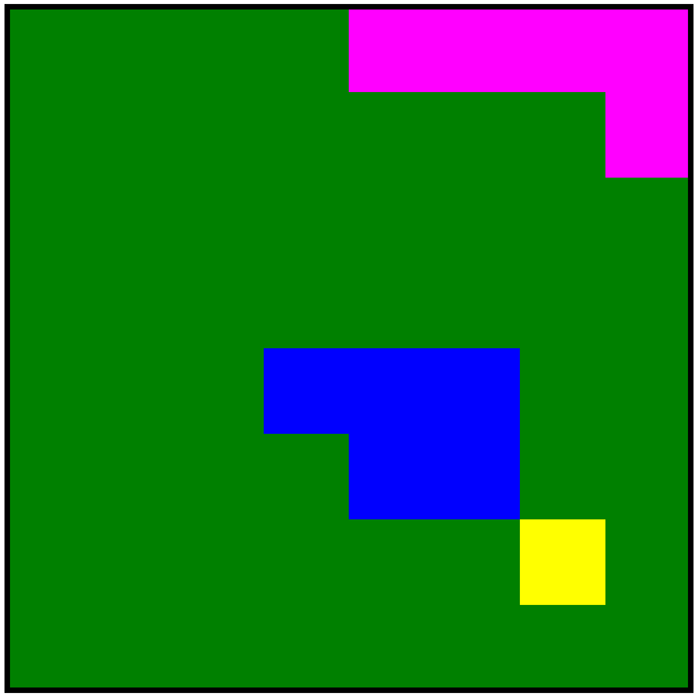




Quadtree (split+merge)

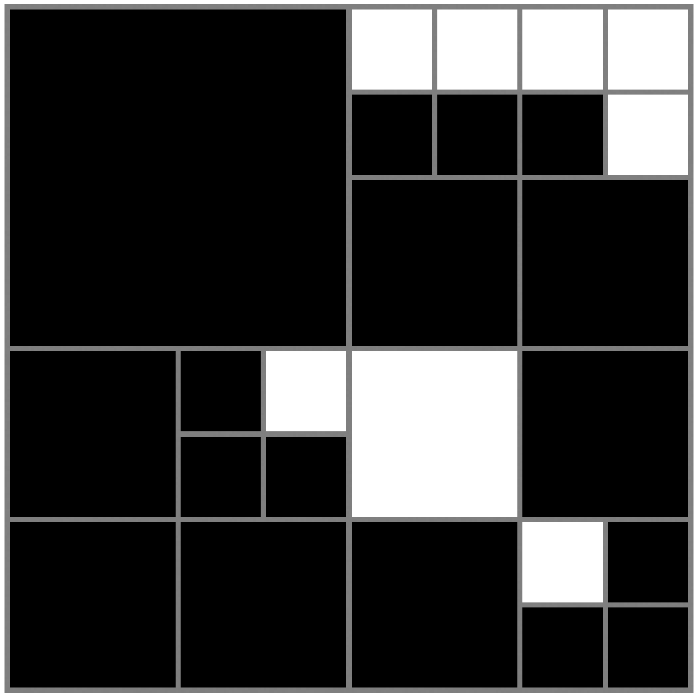
I quadtree possono essere usati per la segmentazione: si fondono le regioni 4-adiacenti con medesima proprietà *P*.

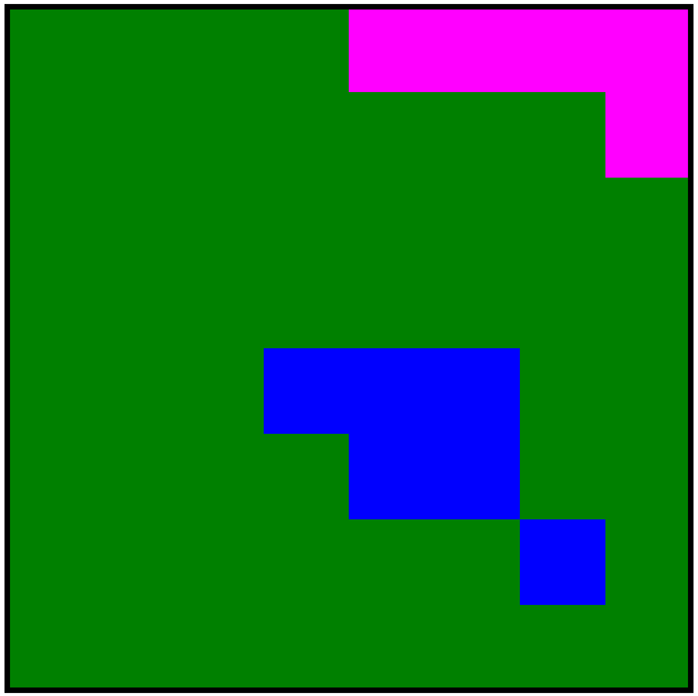




Quadtree (split+merge)

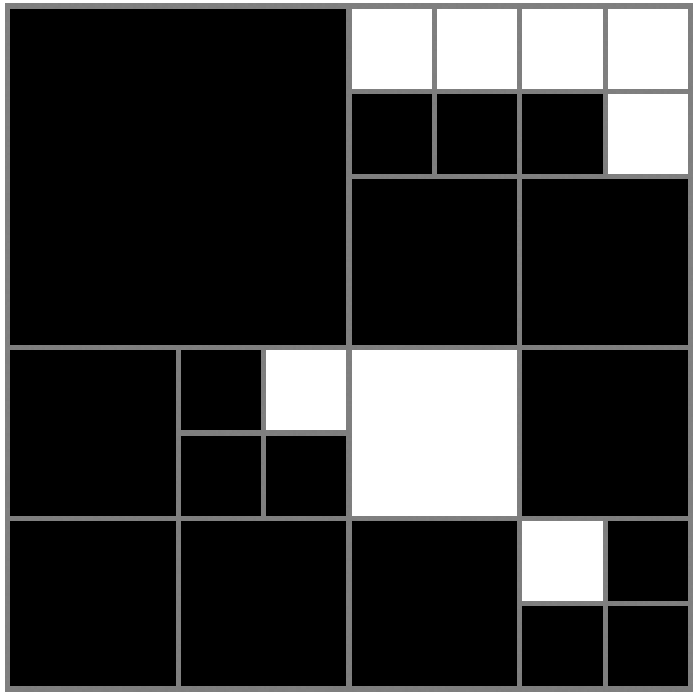
I quadtree possono essere usati per la segmentazione: si fondono le regioni 8-adiacenti con medesima proprietà *P*.



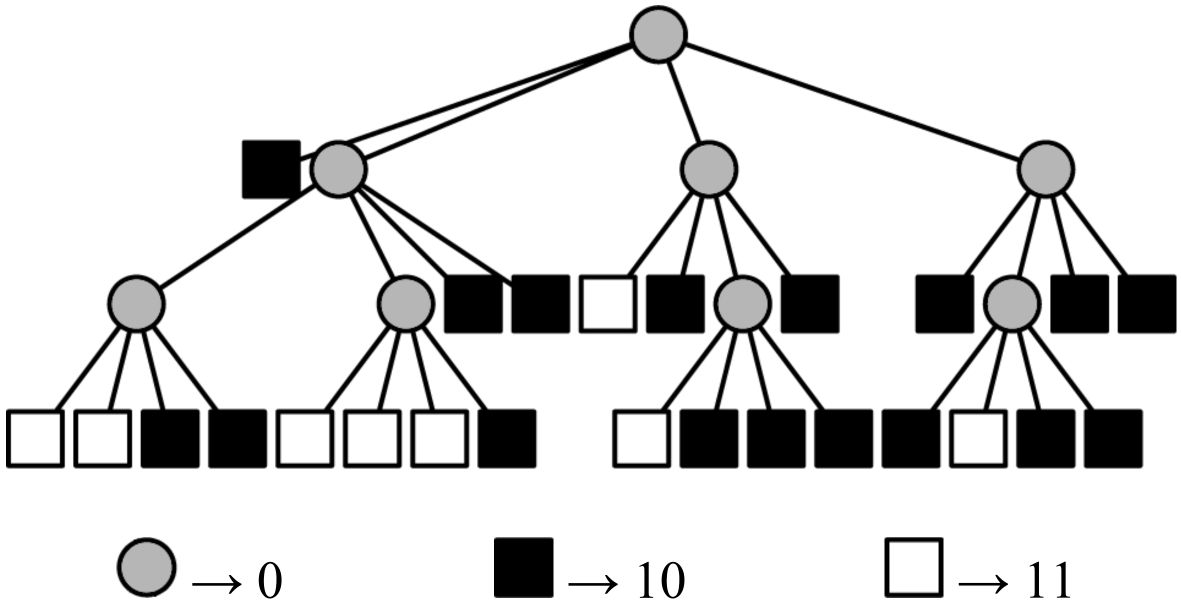


Quadtree (compressione immagini binarie)

I quadtree possono essere usati per rappresentare le immagini binarie con un codice in genere non particolarmente efficiente...



64 bit (raw)

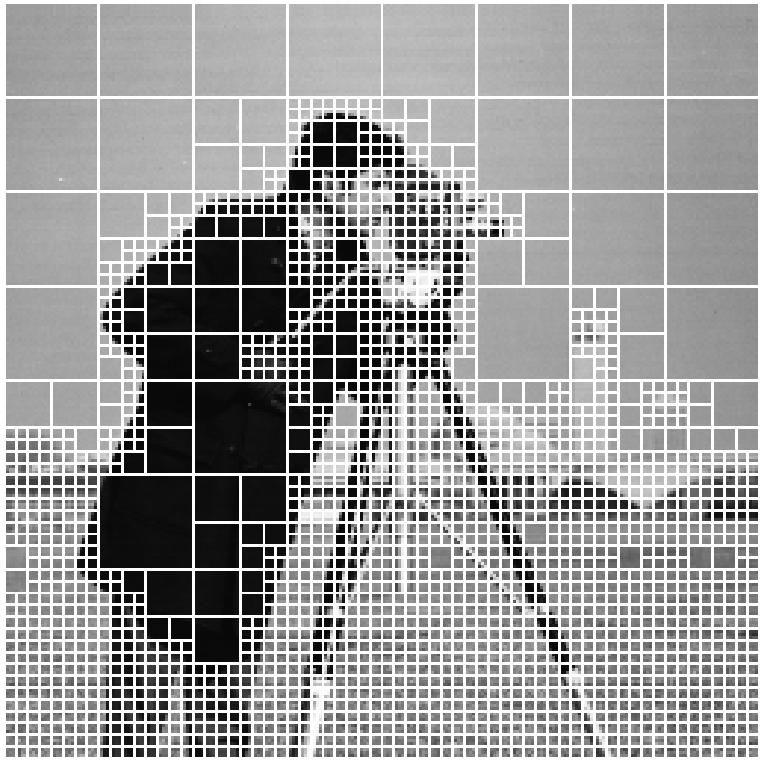


58 bit (quadtree)

BFS:0 10 0 0 0 0 0 10 10 11 10 0 10 10 0 10 10 11 11 10 10 11 11 11 10 11 10 10 10 10 11 10 10

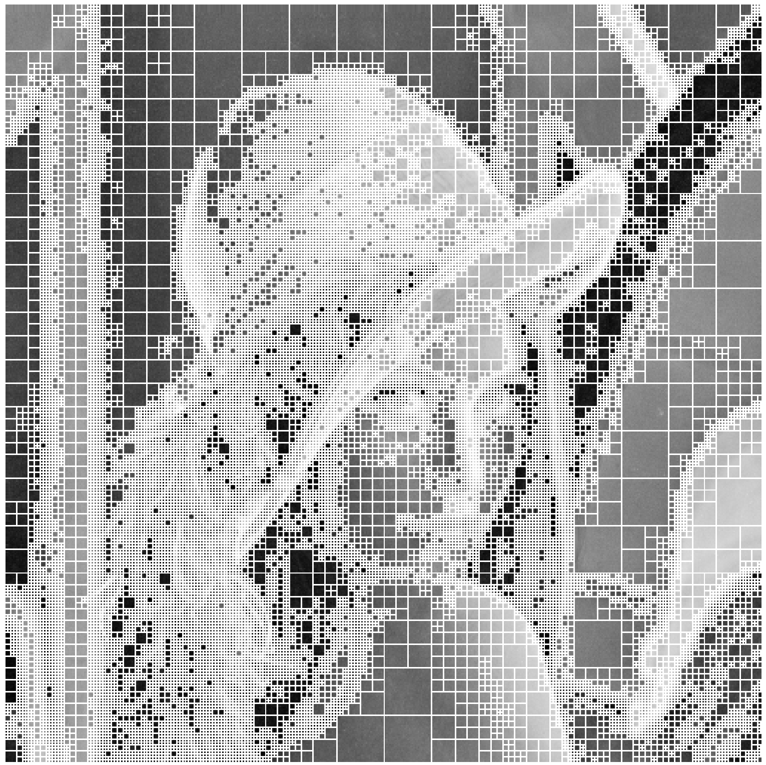
Quadtree (compressione)

Per estendere i quadtree alle immagini con *G* livelli di grigio, si definisce un *criterio di uniformità*, basato sulla varianza dei valori dei pixel nel medesimo blocco: se superiore ad una data soglia, il blocco deve essere suddiviso.



Quadtree (compressione)

Per estendere i quadtree alle immagini con G livelli di grigio, si definisce un *criterio di uniformità*, basato sulla varianza dei valori dei pixel nel medesimo blocco: se superiore ad una data soglia, il blocco deve essere suddiviso.



Quadtree (compressione)

Lena

quatree ($t_\sigma = 5.00$)

Quadtree (compressione)

Lena

quatree ($t_\sigma = 10.00$)

Quadtree (compressione)

Lena

quatree ($t_\sigma = 20.00$)