

Homework 1: codifica lossless d'immagini

OBIETTIVO

L'obiettivo di questo HW è quello d'implementare una semplice tecnica di codifica lossless d'immagini, basata su predizione spaziale e codifica Exp-Golomb (EG). Prima di effettuare la codifica, si richiede uno studio dell'entropia dell'immagine e dell'errore di predizione, in modo molto simile a quello presentato nella demo di codifica lossless presente sul moodle.

Per realizzare questo HW bisogna effettuare degli esperimenti di codifica, eventualmente produrre del codice ed infine scrivere un report che illustri i. Il codice può essere scritto in qualsiasi linguaggio, ma si consiglia Python (notebook) o Matlab (Live Script). Bisognerà consegnare **unicamente il report**.

ESPERIMENTI

Parte 1 (un punto)

Per la prima **parte è possibile riutilizzare in tutto o in parte il codice fornito nella demo su codifica lossless**. Il codice, sia esso in Matlab, Python, JAVA, oppure C/C++, deve eseguire le seguenti operazioni:

1. Caricare un'immagine a livelli di grigio oppure a colori, ma in tal caso bisogna estrarne la componente di luminanza, che può essere approssimata come media delle componenti RGB.
2. Sia $f(n, m)$ l'immagine a livelli di grigio: stimarne l'entropia esprimendola in bit per pixel
3. Utilizzare un'applicazione come zip in Windows oppure gzip in Linux e calcolare il bitrate risultante (dimensione del file in bit diviso numero di pixel)
4. Confrontare l'entropia ottenuta al punto 1 e il tasso ottenuto al punto 3. Discutere il risultato
5. Effettuare la codifica predittiva "**semplice**"
 - 5.1. L'immagine è rappresentata, dopo una scansione riga per riga (*raster scan*), su un **vettore** x
 - 5.2. La predizione di $x(n)$ è $x(n-1)$, tranne per il primo pixel per il quale la predizione è 128
 - 5.3. L'errore di predizione è $y(n) = x(n) - x(n-1)$, tranne per il primo pixel per il quale $y(0) = x(0) - 128$ [in Matlab, $y(1) = x(1) - 128$].
6. Stimare l'entropia dell'errore di predizione y
7. Valutare il *numero di bit necessari* per codificare l'errore di predizione y con la codifica *Exp Golomb con segno*, dedurre il bitrate di codifica e confrontare tale valore con quello ottenuto ai punti 1, 3 e 5
8. Ripetere gli esperimenti per più immagini e riportare i risultati. Commentare quanto trovato.

Parte 2 (un punto)

Per la seconda parte è possibile riutilizzare qualsiasi codice, in particolare per la codifica Exp-Golomb e per la stima dell'entropia. Bisognerà effettuare le operazioni seguenti:

1. Effettuare la codifica predittiva "**avanzata**": per prima cosa si costruisce il predittore p tramite una scansione dell'immagine f :
 - 1.1. Per ogni pixel dell'immagine in posizione n, m
 - 1.2. Se è il primo pixel, il predittore è $p(0,0) = f(0,0) - 128$ [In Matlab, $p(1,1) = f(1,1) - 128$
 - 1.3. Altrimenti, se siamo sulla prima riga, il predittore è il pixel a sinistra di quello corrente

- 1.4. Altrimenti, se siamo sulla prima colonna, il predittore è quello in alto
- 1.5. Altrimenti, se siamo sull'ultima colonna, il predittore è il valore mediano tra $f(n-1, m)$, $f(n, m-1)$, e $f(n-1, m-1)$.
- 1.6. Altrimenti il predittore è il valore mediano tra $f(n-1, m)$, $f(n, m-1)$, e $f(n-1, m+1)$.
- 1.7. Una volta costruito il predittore, si calcola l'errore di predizione $y = f - p$ (è un immagine)
2. Valutare l'entropia di y
3. Valutare il numero di bit necessari per codificare l'errore di predizione con la codifica Exp Golomb con segno e dedurre il tasso di codifica
4. Confrontare l'entropia e il tasso di codifica ottenuti con la codifica predittiva avanzata con quelle del caso "semplice", con l'entropia dell'immagine e con il tasso di codifica dell'applicazione zip
5. Ripetere gli esperimenti per più immagini e riportare i risultati. Commentare quanto trovato.

REPORT

Che cosa mettere nel report per la prima parte. I risultati sperimentali (immagini e i 4 valori calcolati: entropia, tasso zip, entropia errore, tasso codifica signed EG) e la discussione di tali risultati. Non è richiesto per questa prima parte scrivere codice: si può usare quello della demo. Inoltre, nel report non è necessario inserire il codice per questa prima parte. La prima parte sarà considerata eseguita se nel report ci sono i risultati sperimentali commentati in modo corretto.

Che cosa mettere nel report per la seconda parte. Il codice che implementa il predittore avanzato; i risultati (entropia e tasso di codifica con signed EG); il commento a tali risultati. Per considerare eseguita la seconda parte, il codice deve funzionare come richiesto e i risultati devono essere commentati correttamente.

Inviare unicamente il file PDF. Il sistema non accetterà altri file.

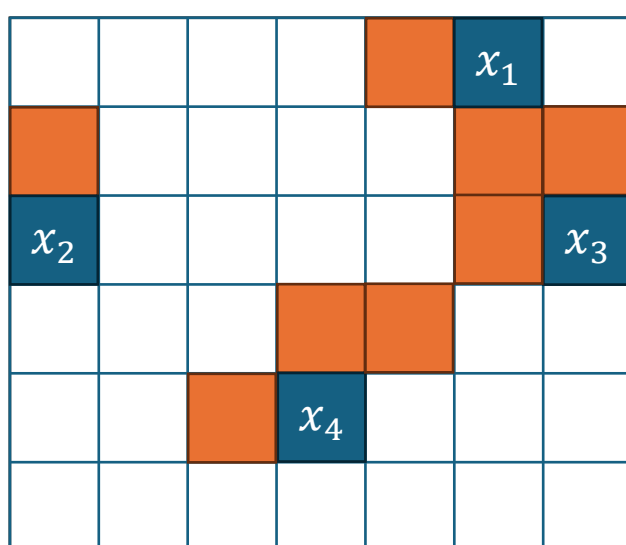


Illustrazione della predizione "avanzata". Sulla prima riga (pixel x_1) il predittore è il vicino di sinistra. Sulla prima colonna (pixel x_2) il predittore è il vicino in alto. Se il pixel è "al centro" (cioè né sulla prima riga, né sulla prima colonna) il predittore è il valore mediano tra i tre vicini (sinistra, alto, alto a destra). Si noti che sull'ultima colonna non esiste un valore in alto **a destra** quindi esso è sostituito dal valore del vicino in alto a sx.