

RIMOZIONE DI "BLOCKING-ARTIFACTS" DA IMMAGINI CODIFICATE TRAMITE DCT

di Andrea Coslovich e Daniele Palazzolo

[Documento pubblicato sulla **Andrea Coslovich HomePage**: <http://sites.google.com/site/andreacoslovich/>]

1. Introduzione:

Nelle immagini compresse con le tecniche che si basano sulla quantizzazione dei coefficienti DCT, specialmente per rapporti di compressione piuttosto alti, si possono individuare senza alcuna difficoltà i bordi dei blocchi usati per la compressione. Questo fenomeno, detto "blocking-artifacts" è spiacevole da vedere e quindi si deve per lo meno tentare la sua eliminazione. Lo scopo di questo lavoro è studiare alcune tecniche per la sua riduzione.

2. Descrizione dei metodi:

2.1. "Blocking-artifacts":

Osservando un'immagine compressa in modo molto spinto (ovvero con un rapporto di compressione elevato) con la DCT si nota subito la presenza del fastidioso fenomeno di blocchettatura consistente nella visibilità dei bordi dei blocchi utilizzati in fase di compressione. Il fenomeno è tanto più marcato quanto più alto è il rapporto di compressione usato. Questo problema è dovuto alla non perfetta corrispondenza tra i margini di blocchi che sono stati processati separatamente dall'algoritmo di compressione.



Esempio di immagine affetta da "blocking-artifacts"

Per risolvere questo problema si possono adottare due approcci antitetici, ovvero:

- Utilizzare delle tecniche di compressione ottimizzate per la riduzione del blocking-artifacts;
- Applicare dei processi di miglioramento della qualità alle immagini ricostruite.

Il primo metodo pur garantendo in taluni casi ottimi risultati è spesso ritenuto poco efficiente a causa del notevole aumento della complessità degli algoritmi di codifica e decodifica. Il secondo, invece, è ritenuto migliore poiché ha il vantaggio di garantire solitamente minore complessità computazionale e di non richiedere modifiche almeno per quanto riguarda l'apparecchiatura (o il software) che effettua la compressione. Bisogna tenere conto, infatti, che l'uso

di compressioni particolarmente drastiche si ritrova spesso in applicazioni concernenti la trasmissione di sequenze video, in cui l'elaborazione può dover essere fatta in tempo reale con la necessità, quindi, di utilizzare algoritmi veloci. Il presente lavoro tiene conto di queste osservazioni e analizza quindi le caratteristiche dei metodi di "elaborazione a posteriori".

2.2. Metodi per la riduzione a posteriori del "blocking-artifacts":

Un'accurata analisi di un'immagine affetta da "blocking-artifacts" permette di osservare che tale problema consiste sostanzialmente in un'introduzione nell'immagine di componenti di alta frequenza consistenti in bruschi "gradini" di luminosità al passaggio tra due blocchi adiacenti. Si potrebbe quindi pensare, in prima analisi, di trattare le immagini ricostruite con un semplice filtro passa basso. In realtà nella progettazione di una tecnica per la riduzione della bloccettatura bisogna cercare di centrare almeno tre obiettivi:

- Buona riduzione degli effetti indesiderati;
- Conservazione del contenuto informativo dell'immagine e delle sue componenti ad alta frequenza;
- Semplicità di implementazione e velocità di esecuzione (in particolare quando la compressione è impiegata per la trasmissione di sequenze video).

Per ottenere queste caratteristiche l'impiego di un comune filtro passa basso non può essere preso in considerazione. Al suo posto possono venir utilizzati degli algoritmi che, sfruttando la conoscenza delle dimensioni dei blocchi usati in fase di codifica, concentrano la loro azione solamente nelle zone realmente interessate dal fenomeno. In questo modo si ottiene contemporaneamente una riduzione della complessità computazionale e un minor decadimento della qualità dell'immagine. Tra le molte possibili tecniche basate su questi concetti ne abbiamo analizzate tre confrontandone le caratteristiche.

Nei prossimi paragrafi sono descritti i metodi di debloccettatura provati.

2.2.1. Filtro passa basso:

Dovendo eliminare un fenomeno di disturbo costituito principalmente da componenti ad alta frequenza la prima idea, come si è detto, non può che essere quella di utilizzare un filtro passa basso. Per semplicità si pensa, naturalmente alla realizzazione di un filtro FIR.

La progettazione di tale filtro può venir effettuata con varie tecniche, nel nostro caso si è adottata quella della "funzione finestra 2D a simmetria circolare", in particolare con una finestra di Hamming. Il progetto si è basato quindi sul seguente schema:

a.) Definizione delle specifiche di risposta in frequenza $H(e^{j\omega_1}, e^{j\omega_2})$. È stata scelta sperimentalmente una frequenza di taglio di $\pi/2$ (che garantisce una certa riduzione del "blocking-effect" senza smorzare troppo i dettagli dell'immagine) e una simmetria di tipo circolare.

b.) Campionamento in frequenza della $H(e^{j\omega_1}, e^{j\omega_2})$ per la determinazione dei coefficienti DFT:

$$H(k_1, k_2) = H\left(e^{j\frac{2\pi}{N_1}k_1}, e^{j\frac{2\pi}{N_2}k_2}\right)$$

Si è osservato che per ottenere buoni risultati è necessario utilizzare almeno $N_1=N_2=5$.

c.) Calcolo della risposta impulsiva $h(n_1, n_2)$ mediante IDFT a partire dai coefficienti $H(k_1, k_2)$.

d.) Calcolo della funzione finestra di Hamming 2D a partire dalla finestra di Hamming 1D utilizzando la relazione:

$$w_2(n_1, n_2) = w_1(\sqrt{n_1^2 + n_2^2})$$

Dove w_1 è la finestra di Hamming in 1D intesa come funzione continua, mentre w_2 è la finestra di Hamming 2D a simmetria circolare che viene calcolata.

e.) Calcolo della risposta impulsiva modificata con:

$$h'(n_1, n_2) = h(n_1, n_2) \times w_2(n_1, n_2)$$

Fortunatamente questo procedimento può essere svolto molto semplicemente ed in modo quasi del tutto automatico utilizzando le funzioni messe a disposizione dall'"Image Processing Toolbox" di MatLab.

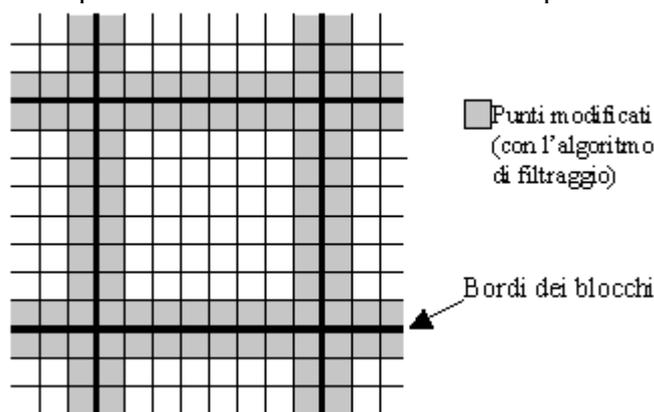
Il filtro va poi applicato all'immagine mediante la convoluzione 2D:

$$y(n_1, n_2) = \sum_{m_1=0}^{M_1} \sum_{m_2=0}^{M_2} h(m_1, m_2) x(n_1 - m_1, n_2 - m_2)$$

2.2.2. Filtro passa basso solo sui bordi dei blocchi:

La prima idea per sfruttare le informazioni sulle dimensioni della blocchettatura al fine di ottenere una migliore qualità dell'immagine rielaborata e di ridurre la complessità computazionale è quella di applicare il filtro passa basso solo sui bordi dei blocchi utilizzati in fase di compressione.

Questa tecnica può venir implementata molto semplicemente eseguendo le operazioni legate alla convoluzione non per tutti i punti dell'immagine ma solo per quelli nei pressi del "confine" tra due blocchi. Nel nostro caso si è scelto di calcolare mediante l'algoritmo di filtraggio soltanto un punto prima del bordo del blocco ed un punto dopo il bordo del blocco, lasciando invariati tutti gli altri punti. Graficamente la situazione si può riassumere come segue:



In altre parole la matrice che descrive l'immagine rielaborata è definita nel seguente modo:

$$y(n_1, n_2) = \begin{cases} \sum_{m_1=0}^{M_1} \sum_{m_2=0}^{M_2} h(m_1, m_2) x(n_1 - m_1, n_2 - m_2) & \text{sui bordi} \\ x(n_1, n_2) & \text{altrove} \end{cases}$$

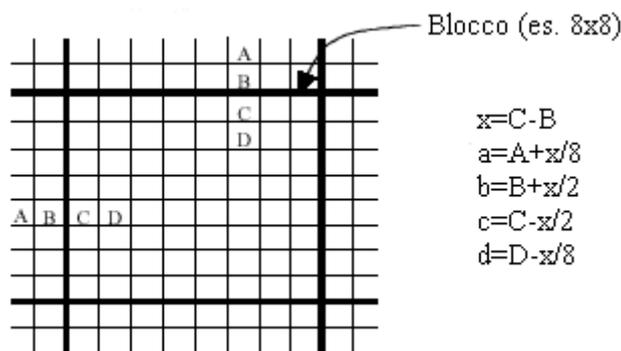
2.2.3. Algoritmo per la deblocchettatura (deblocking algorithm):

Per ridurre ancora la complessità computazionale della tecnica di rielaborazione dell'immagine è necessario ricorrere ad un algoritmo del tutto diverso studiato appositamente per risolvere il problema.

Anche in questo caso si possono adottare diverse soluzioni aventi tutte i loro vantaggi e svantaggi. Nel nostro caso si è scelto ancora una volta di ritenere fondamentali la semplicità di implementazione e la velocità di esecuzione dell'algoritmo. In quest'ottica si è deciso di utilizzare un metodo basato sul lavoro di due ricercatori indiani della "Pace Soft Silicon" e "Sasken Communication" (vedi [bibliografia](#)) che ha proprio nella semplicità il suo punto di forza.

L'idea che sta alla base di questa tecnica è quella di individuare i bordi dei blocchi e modificare i punti nei pressi di tali bordi in modo da trasformare i repentini cambi di luminosità presenti in variazioni più graduali.

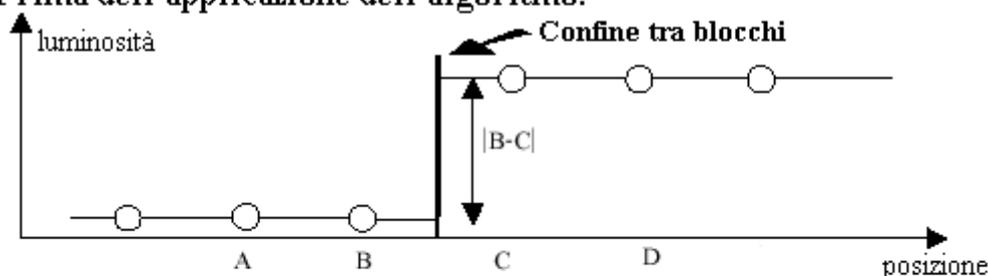
In sostanza l'elaborazione modifica quattro pixel a cavallo dei bordi sia orizzontali che verticali dei blocchi secondo lo schema seguente:



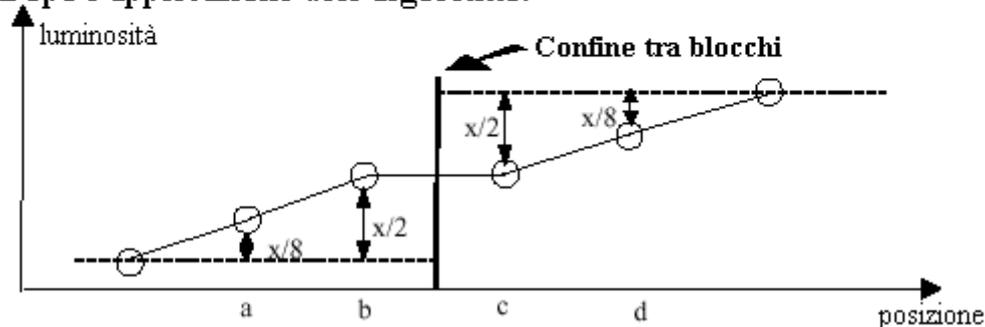
Dove A, B, C e D rappresentano i valori di luminosità dei pixel prima di applicare il metodo, mentre a, b, c e d rappresentano i valori relativi ai medesimi punti dopo l'applicazione del metodo.

Ciò che accade con l'applicazione dell'algorithm può essere ben evidenziato mediante i seguenti grafici in cui viene riportato l'esempio di una possibile situazione nei pressi del "confine" tra due blocchi:

Prima dell'applicazione dell'algorithm:



Dopo l'applicazione dell'algorithm:

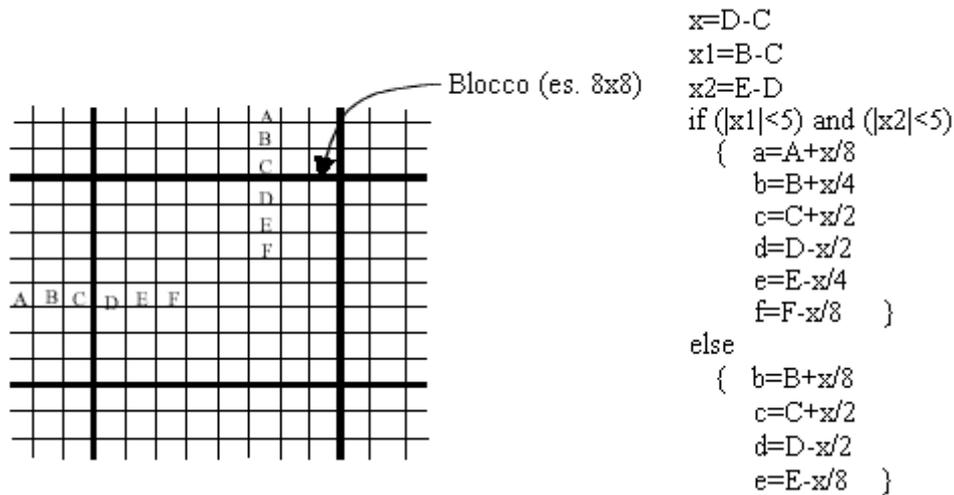


2.2.4. Algorithmo per la deblocchettatura adattativo (adaptive deblocking algorithm):

Con l'algorithm precedente si effettuava un filtraggio su soli quattro punti. Questa tecnica si può definire "leggera" poichè modifica poco l'immagine garantendo in questo modo una buona conservazione dei dettagli. In certi casi però può essere utile agire in modo più deciso per eliminare "blocking-artifacts" molto forti.

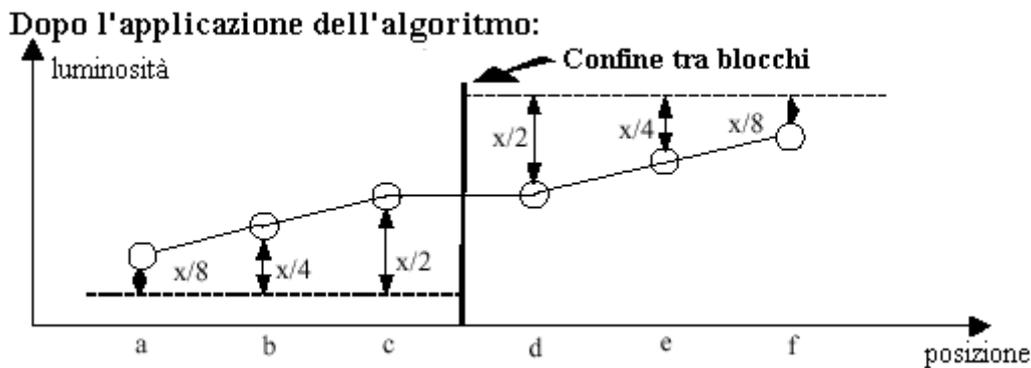
Per conciliare questa esigenza con quella della conservazione dei dettagli gli ideatori dell'algorithm precedente ne propongono una versione modificata che comprende un meccanismo adattativo basato sulla determinazione della quantità di dettagli presenti nell'immagine nei pressi dei bordi dei blocchi. Questa misura viene effettuata valutando le differenze di luminosità tra gli ultimi due pixel all'interno di un blocco e tra i primi due pixel del blocco successivo.

Più grandi sono queste differenze, maggiore è la quantità di dettagli presenti. In pratica si introdurrà, quindi, una soglia il cui superamento comporterà l'applicazione del filtro corto (quello dell'algorithm precedente) mentre per i restanti casi verrà applicato un filtro più lungo (operante su sei punti). Schematicamente la tecnica utilizzata è descrivibile come segue:



Dove A, B, C, D, E ed F rappresentano i valori di luminosità dei pixel prima di applicare il metodo, mentre a, b, c, d, e ed f rappresentano i valori relativi ai medesimi punti dopo l'applicazione del metodo.

Con un semplice grafico è possibile mettere in evidenza cosa accade con l'applicazione del filtro più lungo e permettere, quindi, un confronto con il caso precedente:



2.3. Prove effettuate:

Tutti i metodi presi in considerazione e descritti nei paragrafi precedenti sono stati provati per capire quali sono le loro caratteristiche e soprattutto per stabilire come essi si comportano al variare del rapporto di compressione usato. Si è voluto, cioè, determinare oltre quale valore del rapporto di compressione è davvero conveniente l'uso delle tecniche analizzate.

Nelle prove è stato utilizzato un algorithm di compressione da noi sviluppato per creare le immagini "corrotte" mentre le valutazioni quantitative sulla qualità dell'immagine sono state effettuate con il PSNR e con un metodo di misurazione del "blocking-artifacts" basato su un lavoro di tre ricercatori della "University of Texas" (vedi [bibliografia](#)). Nei prossimi paragrafi verranno date indicazioni più precise su questi tre punti.

2.3.1. Algorithm di compressione:

Per effettuare le prove le tecniche analizzate sono state applicate a delle immagini "comprese" con un metodo da noi sviluppato basato sulla quantizzazione dei coefficienti della DCT e sull'eliminazione di una certa quantità di essi. In sostanza l'algorithm di codifica usato presenta il seguente schema operativo:

- a.) Divisione dell'immagine in blocchi di dimensione $N \times N$, dove N può venir scelto a piacere (in realtà per le prove è stato sempre usato il valore $N=8$ perchè è quello normalmente utilizzato in tutti gli algoritmi di compressione reali).
- b.) Applicazione della DCT bidimensionale ad ogni blocco con determinazione di una matrice di coefficienti di dimensione $N \times N$.

- c.) Quantizzazione dei coefficienti con codifica di ciascuno di essi mediante un numero b di bit selezionabile a piacere. In particolare la quantizzazione viene fatta calcolando il massimo (M) ed il minimo (m) tra tutti i coefficienti di una determinata posizione all'interno dei blocchi $N \times N$ e utilizzando come passo di quantizzazione il valore: $(M-m)/2^b$.
- d.) Eliminazione dei coefficienti DCT non contenuti in quadrato di lato $S \times S$, dove S può essere scelto a piacere. La decodifica avviene sostanzialmente in modo contrario, ovvero con la seguente procedura:
- Ricostruzione dei coefficienti della DCT di ogni blocco a partire dai valori codificati. Questo passaggio può avvenire solamente grazie alla conoscenza dei valori di minimo (m) e di massimo (M) relativi a ciascuna posizione all'interno delle matrici ($N \times N$) dei coefficienti.
 - Calcolo dei valori di luminosità dei pixel di ciascun blocco dell'immagine mediante la IDCT bidimensionale.
 - Ricostruzione dell'immagine originale per giustapposizione dei blocchi ricalcolati al passo precedente.
- Questo metodo di codifica è stato scelto perchè è molto semplice e pur non ricalcando in modo fedele alcun tipo di algoritmo di compressione "reale" presenta alcune analogie con il comunissimo standard JPEG.

2.3.2. PSNR:

Il PSNR è un metodo che permette di misurare la qualità di un'immagine rumorosa (o corrotta) mediante una comparazione con l'immagine pulita di partenza. Il PSNR si calcola come una sorta di rapporto segnale/rumore (espresso in decibel) in cui la valutazione del rumore è data dall'*mse* (mean square error) mentre per il segnale si usa una stima di massima considerando l'immagine completamente bianca. Si ha in pratica:

$$mse = \sqrt{\frac{\sum_i \sum_j (x_{or}(i,j) - x(i,j))^2}{N_1 \times N_2}}$$

$$PSNR = 10 \log_{10} \left(\frac{255^2 \times N_1 \times N_2}{mse^2} \right)$$

dove x è la matrice rappresentante l'immagine oggetto di misura, x_{or} è la matrice rappresentante l'immagine originale, mentre N_1 ed N_2 sono le dimensioni orizzontale e verticale delle immagini.

Questa misura sebbene sia universalmente utilizzata per la valutazione della qualità delle immagini risultanti da processi di elaborazione in effetti non può fornire alcuna informazione specifica sulla gravità del "blocking-effect" dovuto alla compressione. Per questo motivo si rende necessario affiancarle una misurazione mirata di questo fenomeno ed una valutazione qualitativa della bontà delle immagini.

2.3.3. Misura del blocking-artifacts:

Per la misurazione del "blocking-artifacts" (MBA) è stato scelto il metodo "blind measurement of blocking artifacts" (abbreviato d'ora in poi con BMBA) proposto da tre ricercatori dell'università del Texas (vedi [bibliografia](#)). Il metodo BMBA suggerisce di considerare l'immagine affetta da "blocking-artifacts" come composta da due parti: una di sola blocchettatura e una costituita dall'immagine non corrotta. In quest'ottica, quindi, la misurazione può essere basata sulla differenza di potenza tra queste due componenti. In particolare il metodo permette la misurazione anche in assenza dell'immagine originale grazie all'approssimazione dello spettro in potenza di quest'ultima con lo spettro (in potenza) dell'immagine corrotta filtrato con un filtro mediano.

L'algoritmo BMBA prevede i seguenti punti da applicare sia procedendo per righe che per colonne:

- Creazione a partire dall'immagine da testare (x) di una matrice delle differenze (g) contenente i valori assoluti delle differenze tra due punti successivi di x . Ad esempio la matrice delle differenze orizzontali (per la misura del blocking-effect verticale) sarà definita come: $g(i,j) = |x(i,j) - x(i,j-1)|$.
- Inserimento dei valori della matrice delle differenze in un vettore. Per l'MBA verticale: $s(iM+j) = g(i,j)$ per $0 \leq i, j \leq M-1$ dove M rappresenta la dimensione orizzontale dell'immagine.
- Calcolo dello spettro in potenza di s . Per questa operazione viene effettuata la FFT di suoi segmenti lunghi N (con N che dev'essere una potenza del 2, nel nostro caso $N=256$), poi il calcolo dello spettro in potenza di ciascun segmento con la relazione:

$$P^{(k)}(l) = \begin{cases} 2|B^{(k)}(l)|^2 & 1 \leq l \leq N/2 - 1 \\ |B^{(k)}(l)|^2 & l = 0, N/2 \end{cases}$$

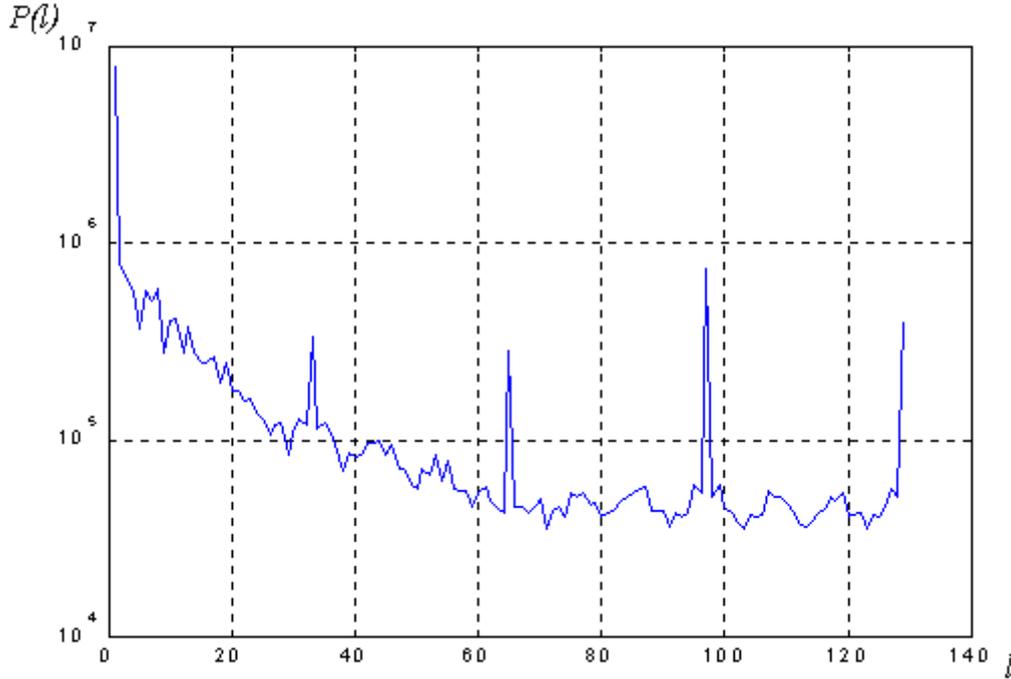
dove $B^{(k)}$ rappresenta lo spettro del k -esimo segmento di s , mentre $P^{(k)}$ è il suo spettro in potenza. Infine viene calcolata la media degli spettri di potenza dei segmenti per determinare lo spettro in potenza medio di s :

$$P(l) = \frac{1}{L} \sum_{k=1}^L P^{(k)}(l) \quad 0 \leq l \leq N/2 \quad \text{dove } L \text{ è il numero di segmenti in cui è stato diviso } s.$$

Se l'immagine x analizzata presenta fenomeni di blocchettatura lo spettro $P(l)$ presenta dei picchi nelle posizioni:

$$n \frac{N}{d} \quad 0 \leq n \leq d/2 \quad (\text{dove } d = \text{dimensione blocchi})$$

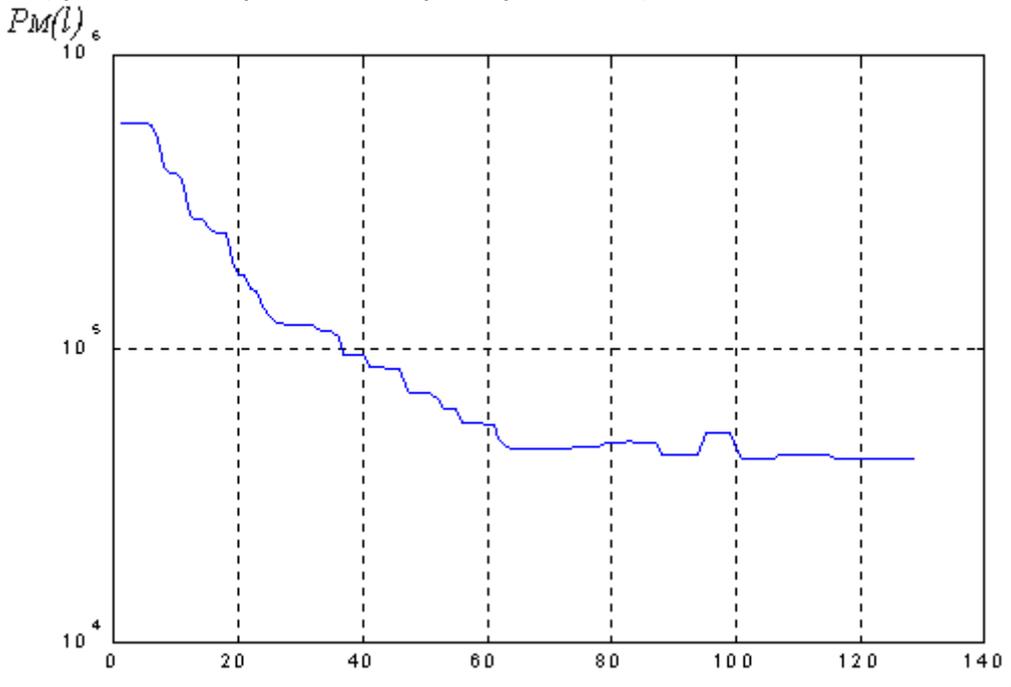
dovuti alle componenti periodiche del segnale introdotte dalla blocchettatura stessa. Ad esempio per l'immagine di Lena compressa utilizzando la DCT su blocchi 8x8 si ottiene uno spettro del seguente tipo:



d.) Calcolo approssimato dello spettro dell'immagine non corrotta ($P_M(l)$) mediante applicazione di un filtro mediano a $P(l)$:

$$P_M(l) = \text{Median}(P(l-K), \dots, P(l), \dots, P(l+K)) \quad (\text{dove } K \text{ dev'essere scelto in relazione a } N, \text{ per noi } K=4)$$

Per l'immagine di Lena, presa ad esempio anche nel passo precedente, si ottiene:



e.) Determinazione della misura di "blocking-artifacts" mediante valutazione delle differenze di potenza degli spettri $P(l)$ e $P_M(l)$ nei punti $l=n(N/d)$. In realtà questo calcolo viene fatto utilizzando la seguente relazione:

$$M = \frac{d}{d-1} \sum_{n=1}^{d/2} P\left(n \frac{N}{d}\right) - P_M\left(n \frac{N}{d}\right)$$

perchè impossibile effettuare una valutazione corretta per $n=0$, valore per il quale ci si pone in una zona dello spettro in cui si concentra gran parte dell'energia di qualsiasi immagine.

Infine, svolti i passi descritti, è necessario calcolare il valore della misura relativa all'intera immagine come media tra la valutazione relativa alle righe (M_o) e quella relativa alle colonne (M_v).

$$M_B = 0.5(M_v + M_o)$$

Le misure così calcolate presentano molto spesso valori molto grandi e quindi per ottenere delle quantità più facilmente maneggiabili si rende necessario esprimerle in termini logaritmici con la seguente relazione:

$$MBA = \begin{cases} \log_{10}(M_B) & M_B > 1 \\ 0 & M_B \leq 1 \end{cases}$$

2.4. Programmi MatLab:

Per permettere l'analisi dei metodi di riduzione del "blocking-artifacts" e le misurazioni necessarie sono stati messi a punto dei semplici programmi in linguaggio MatLab le cui caratteristiche sono descritte di seguito:

RBA: Programma principale. Gestisce l'input/output con l'utente e permette di effettuare le prove sui metodi richiamando gli altri programmi.

DODCT: Accetta in ingresso la matrice delle luminosità dell'immagine originale e i parametri della compressione. Restituisce in uscita l'immagine risultante dal processo di compressione-decompressione (vedi paragrafo [2.3.1](#)) e una stima del rapporto di compressione.

PB1: Accetta in ingresso la matrice delle luminosità dell'immagine da filtrare e i parametri del filtro. Restituisce in uscita l'immagine filtrata (vedi paragrafo [2.2.1](#)).

PB2: Accetta in ingresso la matrice delle luminosità dell'immagine da filtrare, i parametri del filtro e la dimensione dei blocchi. Restituisce in uscita l'immagine filtrata solo sui bordi dei blocchi (vedi paragrafo [2.2.2](#)).

DA1: Accetta in ingresso la matrice delle luminosità dell'immagine da elaborare e la dimensione dei blocchi. Restituisce in uscita l'immagine rielaborata con il "Deblocking Algorithm" (vedi paragrafo [2.2.3](#)).

DA2: Accetta in ingresso la matrice delle luminosità dell'immagine da elaborare e la dimensione dei blocchi. Restituisce in uscita l'immagine rielaborata con l'"Adaptive Deblocking Algorithm" e il numero di applicazioni del filtro lungo (vedi paragrafo [2.2.4](#)).

BMBA: Accetta in ingresso la matrice delle luminosità dell'immagine da misurare e la dimensione dei blocchi. Fornisce in uscita la misura del "blocking-artifacts" (vedi paragrafo [2.3.3](#)).

Il programma RBA sfrutta, inoltre, le seguenti tre funzioni che sono state trovate già pronte perchè messe a disposizione dal Dott. Marsi come materiale di supporto al corso di "[Elaborazione elettronica di segnali e immagini](#)":

LOAD_IMAGE: Permette di caricare un'immagine in formato IMG.

SAVE_IMAGE: Permette di salvare un'immagine in formato IMG.

PSNR: Calcola il PSNR di un immagine (vedi paragrafo [2.3.2](#)).

[**NOTA BENE:** Per provare questi programmi è possibile scaricare il file compresso [rba.zip](#) che contiene questo testo, tutti i programmi MatLab e le immagini necessarie per le prove.]

3. Risultati conseguiti:

I metodi descritti in precedenza sono stati provati applicandoli a due diverse immagini di test: LENA e BOAT.



Immagine LENA



Immagine BOAT

Per entrambe le immagini sono stati effettuati diversi test variando di volta in volta i parametri dell'algoritmo di compressione e cercando quindi di determinare oltre quali valori di rapporto di compressione sia veramente conveniente l'uso dei metodi analizzati. Per ogni prova e per ogni metodo applicato sono stati registrati i valori di PSNR ed MBA e delle impressioni di tipo qualitativo sulle immagini. Questo lavoro ha portato alla realizzazione delle tabelle riportate a seguire.

Per l'immagine LENA:

d S b	R	PSNR O	MBA O	Q	PSNR1	MBA1	G1	PSNR2	MBA2	G2	PSNR3	MBA3	G3	PSNR4	MBA4	G4	Nlun
8 8 8	1,0000	56,8458	0,0000	O	31,0438	0,0000	P	35,0027	5,0567	P	32,9662	6,0933	P	32,9280	6,0899	P	7226
8 8 6	1,3333	44,6198	0,0000	O	30,9420	0,0000	P	34,0378	5,0378	P	32,7939	6,0983	P	32,7666	6,0950	P	7130
8 8 4	2,0000	32,9307	4,6706	B	29,5276	4,0457	P	31,5949	4,7289	N	30,8068	6,1224	N	30,8811	6,1110	N	6922
8 8 2	4,0000	20,5831	6,6240	I	21,3103	5,9136	-	21,3904	5,6862	-	21,8718	6,2097	-	22,0972	6,1388	-	8875
8 6 6	2,3704	35,0962	2,3703	O	30,5380	0,0000	P	32,5508	4,9904	P	30,9885	6,0507	P	30,9745	6,0456	P	7685
8 6 4	3,5556	31,1528	5,3152	B	29,2361	4,1452	M	30,4682	4,7743	N	29,6404	6,0665	M	29,7084	6,0503	N	7533
8 6 2	7,1111	20,5128	6,6594	I	21,2746	5,9199	-	21,3118	5,7739	-	21,7691	6,1813	-	22,0049	6,0922	-	9263
8 4 6	5,3333	29,9417	5,4457	O	28,8952	0,0000	N	29,5565	4,9783	N	28,3415	5,8836	P	28,3022	5,8740	P	8990
8 4 4	8,0000	28,4214	5,8199	B	27,9863	4,4460	M	28,4467	4,9767	M	27,6285	5,9008	M	27,6578	5,8815	M	8630
8 4 3	10,6667	25,5174	6,1878	B	25,7840	5,1799	M	25,9784	5,2352	M	25,7501	5,9341	M	25,8650	5,8902	M	8783
8 4 2	16,0000	20,3218	6,7172	I	21,0926	5,9491	-	21,0858	5,9101	-	21,4915	6,0692	-	21,7446	5,9382	-	10249
8 2 6	21,3333	24,5119	6,3176	B	24,7841	5,3705	N	24,9357	5,5112	N	24,6429	5,6034	N	24,3173	5,5399	M	13033
8 2 4	32,0000	24,1183	6,4116	B	24,4543	5,5186	M	24,5858	5,6120	M	24,3746	5,6362	M	24,1104	5,5440	M	12925
8 2 3	42,6667	23,0133	6,5303	B	23,4573	5,7191	M	23,5406	5,7704	M	23,5223	5,7097	M	23,4032	5,5506	M	13334
8 2 2	64,0000	19,7187	6,7610	I	20,3646	6,0799	-	20,3748	6,0981	-	20,7515	5,9252	-	20,9299	5,6776	-	12602

Per l'immagine BOAT:

d S b	R	PSNR O	MBA O	Q	PSNR1	MBA1	G1	PSNR2	MBA2	G2	PSNR3	MBA3	G3	PSNR4	MBA4	G4	Nlun
8 8 8	1,0000	56,7398	0,0000	O	28,8235	0,0000	P	32,9599	5,4277	P	31,5786	6,2226	P	31,5498	6,2202	P	6321
8 8 6	1,3333	44,5658	0,0000	O	28,7442	0,0000	P	32,7645	5,4150	P	31,4462	6,2287	P	31,4235	6,2258	P	6223
8 8 4	2,0000	32,6204	4,0958	B	27,7970	0,0000	P	30,4087	5,3282	N	29,7606	6,2671	N	29,7989	6,2600	N	5449
8 8 2	4,0000	19,5401	6,6806	I	20,1212	5,9986	-	20,2014	5,7617	-	20,6299	6,3809	-	20,7848	6,3391	-	6489
8 6 6	2,3704	32,9385	0,0000	O	28,2939	0,0000	P	30,4266	5,3160	P	29,2436	6,1700	P	29,2299	6,1657	P	5723
8 6 4	3,5556	30,0145	5,2474	B	27,4272	0,0000	N	28,9515	5,2252	N	28,1997	6,1989	M	28,2235	6,1900	M	5251
8 6 2	7,1111	19,4511	6,7429	I	20,0717	6,0129	-	20,1098	5,8988	-	20,5005	6,3320	-	20,6695	6,2732	-	7214
8 4 6	5,3333	27,8278	5,6902	O	26,6101	0,0000	N	27,3987	5,3099	N	26,3966	5,9492	N	26,3490	5,9405	P	7040
8 4 4	8,0000	26,7828	5,9470	B	26,0104	0,0000	M	26,6311	5,3170	N	25,8635	5,9705	M	25,8552	5,9556	M	6749
8 4 3	10,6667	24,3091	6,3235	B	24,2866	5,2658	M	24,6158	5,5006	M	24,3560	6,0352	M	24,4300	5,9992	M	7084
8 4 2	16,0000	19,2194	6,8315	I	19,8468	6,0624	-	19,8706	6,0680	-	20,1855	6,1856	-	20,3782	6,0742	-	8971
8 2 6	21,3333	23,1504	6,3854	B	23,1499	5,4607	N	23,3840	5,7273	N	23,1271	5,5861	M	22,8634	5,4481	M	13218
8 2 4	32,0000	22,8487	6,4598	B	22,9113	5,5746	M	23,1207	5,7879	M	22,9179	5,6262	M	22,6984	5,4640	M	13106
8 2 3	42,6667	21,8580	6,6372	B	22,0952	5,8540	M	22,2551	5,9699	M	22,2396	5,7734	M	22,1470	5,5322	M	13166
8 2 2	64,0000	18,5888	6,8776	I	19,0943	6,2143	-	19,1253	6,2663	-	19,4363	6,0296	-	19,6181	5,7678	-	12293

Per entrambe le tabelle si ha la seguente legenda:

d: Dimensione blocco DCT;

S: Dimensione del quadrato di coefficienti DCT salvati;

b: Numero di bit utilizzato per la codifica dei coefficienti DCT;

R: Stima del rapporto di compressione: $R=8d^2/(S^2b)$;

PSNR O e **MBA O**: PNSR e MBA dell'immagine compressa **non** rielaborata;

PSNR1 e **MBA1**: PNSR e MBA dell'immagine compressa e rielaborata con il filtro passa basso;

PSNR2 e **MBA2**: PNSR e MBA dell'immagine compressa e rielaborata con il filtro passa basso solo sui bordi;

PSNR3 e **MBA3**: PNSR e MBA dell'immagine compressa e rielaborata con l'algoritmo di debloccatura;

PSNR4 e **MBA4**: PNSR e MBA dell'immagine compressa e rielaborata con l'algoritmo di debloccatura adattativo;

Nlun: Numero applicazioni del filtro lungo nell'algoritmo di debloccatura adattativo;

Q: Giudizio qualitativo sull'immagine compressa con la seguente scala:

O - Immagine di ottima qualità sostanzialmente priva di difetti evidenti;

B - Immagine di qualità accettabile ma con evidente "blocking-artifacts";

I - Immagine di qualità inaccettabile;

G1, **G2**, **G3** e **G4**: Giudizi qualitativi sui metodo:

P - L'immagine peggiora con l'applicazione del metodo;

N - L'immagine non migliora sensibilmente ma nemmeno peggiora con l'applicazione del metodo;

M - L'immagine migliora con l'applicazione del metodo;

Le prime conclusioni qualitative che si possono trarre dalla semplice osservazione delle tabelle sono le seguenti:

- Per immagini compresse di buona qualità in cui il "blocking-effect" non è molto evidente risulta controproducente l'applicazione di qualsiasi rielaborazione tra quelle analizzate. Ad esempio ecco cosa accade all'immagine di Lena per un rapporto di compressione di circa 1:2 (d=8, S=6, b=6):



Immagine originale

Immagine compressa

Passa basso



Passa basso solo sui bordi

Deblocking Algorithm

Adaptive Deblocking Algorithm

- Per immagini in cui il "blocking-effect" è evidente ma non gravissimo il passa basso e il passa basso applicato solo sui bordi dei blocchi risultano in genere entrambi inadeguati perchè il primo elimina troppi dettagli ancora presenti nell'immagine, mentre il secondo non riesce a correggere i difetti presenti. Per questo caso, invece, i due algoritmi di debloccatura danno risultati non facilmente interpretabili poichè, se da un lato riescono a ridurre il fenomeno di "blocking-artifacts" dall'altro tendono ad eliminare una parte dei dettagli dell'immagine. Ecco cosa accade, ad esempio, all'immagine di Lena compressa con rapporto circa 1:3,5 ($d=8$, $S=6$, $b=4$):



Immagine originale

Immagine compressa

Passa basso



Passa basso solo sui bordi

Deblocking Algorithm

Adaptive Deblocking Algorithm

- Per immagini molto corrotte tutti i metodi garantiscono un miglioramento della qualità dell'immagine, ma, il passa basso elimina solamente in parte il blocking-artifacts ed elimina gran parte dei dettagli ancora presenti, il passa basso applicato solo sui bordi dei blocchi non è in grado di eliminare del tutto soddisfacentemente la bloccettatura mentre i due "deblocking algorithm" danno i risultati migliori. Ad esempio ecco cosa accade all'immagine di Lena per un rapporto di compressione di circa 1:32 ($d=8$, $S=2$, $b=4$):



Immagine originale



Immagine compressa



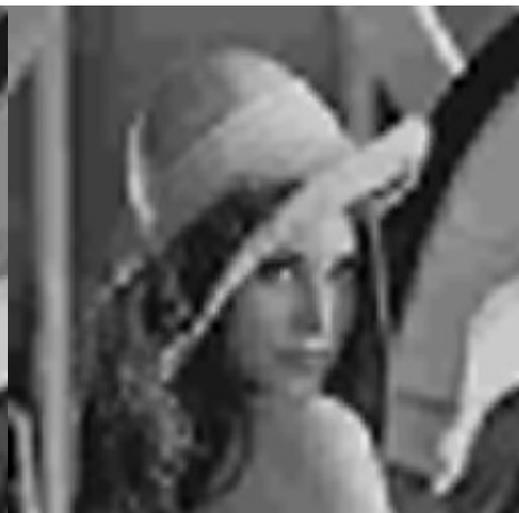
Passa basso



Passa basso solo sui bordi



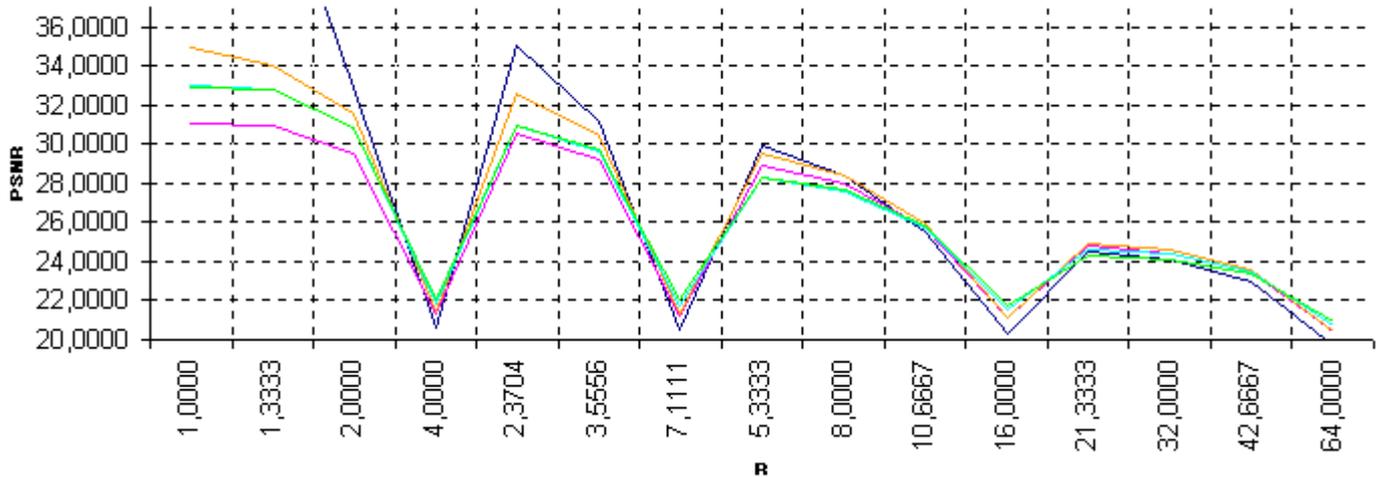
Deblocking Algorithm



Adaptive Deblocking Algorithm

Per ottenere una valutazione quantitativa sul valore del rapporto di compressione oltre il quale è conveniente applicare ciascuno dei metodi analizzati è conveniente tradurre i dati tabellati relativi al PSNR e al MBA in grafici.

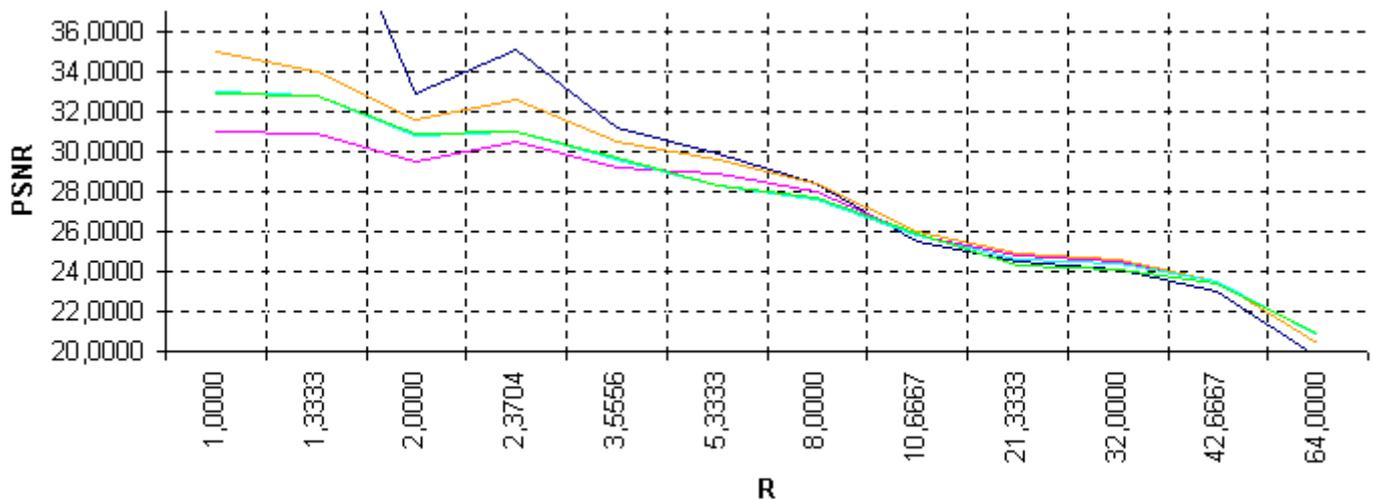
Per l'immagine di **Lena** si ha il seguente grafico del PSNR:



Con: ----- Immagine compressa; ----- Passa Basso;
----- PB solo bordi blocchi; ----- Deblocking Algorithm; ----- Adaptive Deb. Alg.

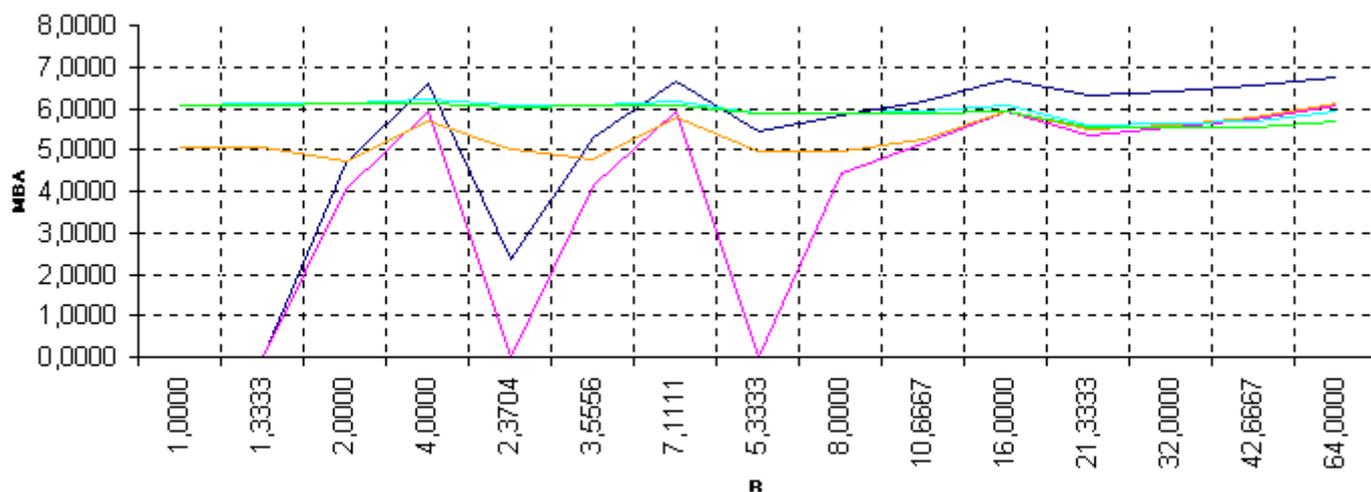
[ATTENZIONE: Le scale dei grafici non sono lineari ed i punti sono riportati nell'ordine in cui compaiono nella tabella]

Di fatto questo grafico è difficile da leggere poichè i picchi verso il basso causati dalle immagini molto degradate impediscono di capire quali curve si trovano più in alto e quali più in basso in ogni punto. Per poter effettuare una lettura migliore è possibile eliminare i punti relativi alle immagini di qualità molto scadente (immagini indicate nella tabella con valore di qualità pari ad **I**) che comunque non sono significativi, ottenendo in questo modo:



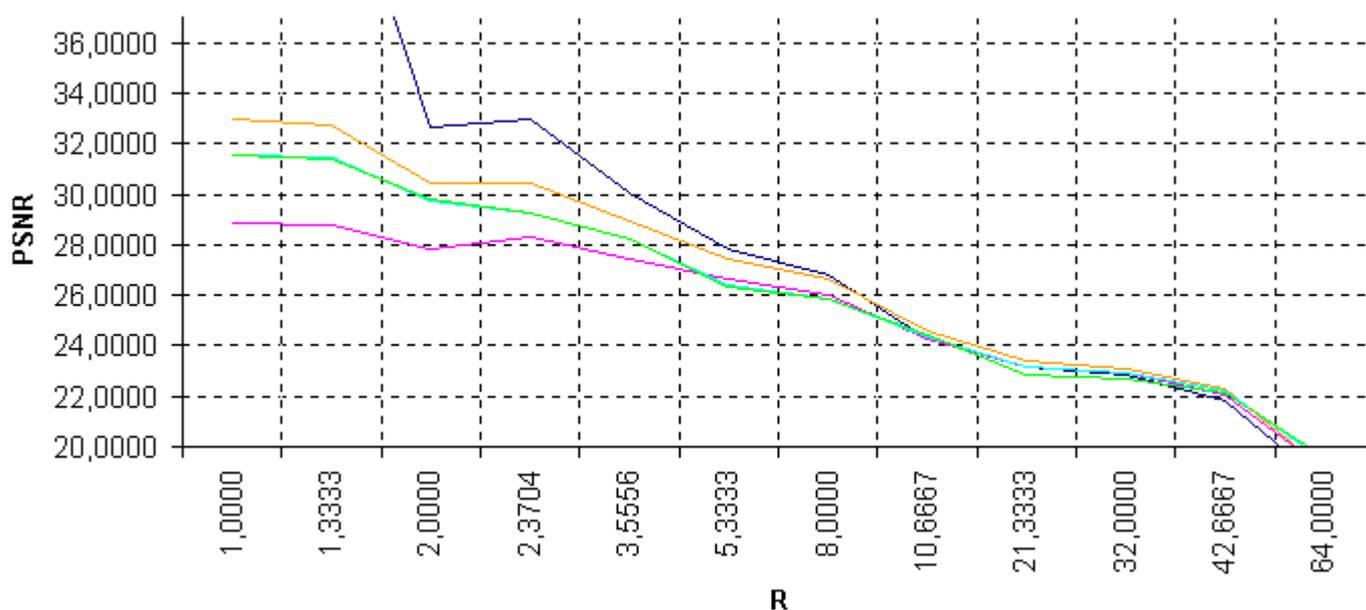
Con: ----- Immagine compressa; ----- Passa Basso;
----- PB solo bordi blocchi; ----- Deblocking Algorithm; ----- Adaptive Deb. Alg.

Per quanto riguarda, invece, la misura del blocking artifacts il grafico è il seguente:

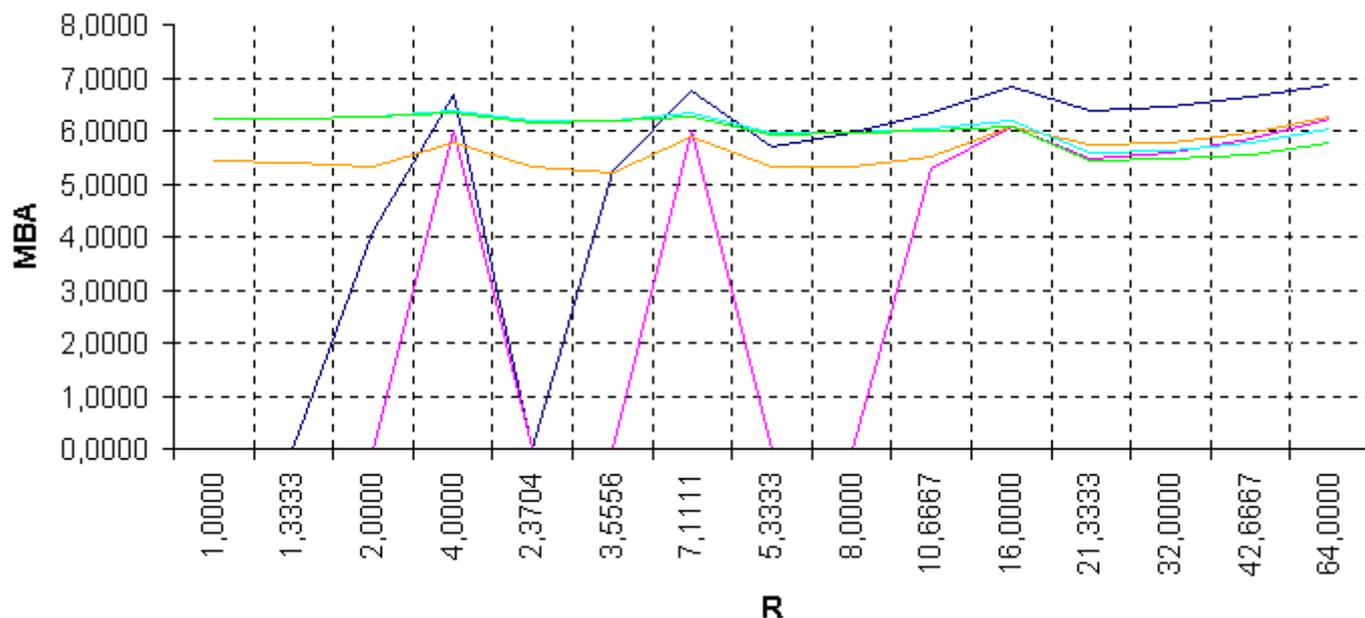


Con: ----- Immagine compressa; ----- Passa Basso;
 ----- PB solo bordi blocchi; ----- Deblocking Algorithm; ----- Adaptive Deb. Alg.

Per l'immagine **Boat** si ottengono i seguenti grafici (quello relativo al PSNR è già depurato dai valori non significativi):



Con: ----- Immagine compressa; ----- Passa Basso;
 ----- PB solo bordi blocchi; ----- Deblocking Algorithm; ----- Adaptive Deb. Alg.



Con: ----- Immagine compressa; ----- Passa Basso;
 ----- PB solo bordi blocchi; ----- Deblocking Algorithm; ----- Adaptive Deb. Alg.

4. Conclusioni:

4.1. Osservazioni generali:

L'analisi effettuata sui quattro metodi presi in considerazione permette di concludere, innanzitutto, che la loro applicazione è plausibile solamente nei casi in cui il rapporto di compressione è piuttosto elevato. Quando ciò non si verifica, infatti, l'applicazione delle tecniche risulta inutile (con conseguente spreco di risorse computazionali) o persino dannosa peggiorando la qualità dell'immagine. La soglia di rapporto di compressione oltre la quale l'applicazione dei metodi è conveniente può venir estrapolata dall'osservazione dei grafici tracciati ed è di circa 1:15. Oltre questo valore l'applicazione delle tecniche comporta un miglioramento sia del PSNR che dell'MBA. In realtà, l'indagine qualitativa può portare a ritoccare questo valore di qualche punto verso il basso fissandolo attorno a 1:10.

4.2. Comparazione dei metodi:

Per effettuare un paragone significativo tra i quattro metodi occorre tenere conto per lo meno di due parametri: la qualità delle immagini ottenute e la complessità computazionale (importante soprattutto in previsione di un loro eventuale utilizzo per l'elaborazione di sequenze video). In quest'ottica si può affermare che i metodi più efficaci risultano i due algoritmi di debloccatura, infatti: il passa basso dà risultati abbastanza soddisfacenti, ma comporta un numero di operazioni molto elevato, dell'ordine di 25 moltiplicazioni e altrettante somme per ogni punto dell'immagine; per il passa basso applicato solo sui bordi il miglioramento della qualità è sostanzialmente analogo ma il costo computazionale è molto minore, occorrono, infatti, soltanto 50 moltiplicazioni e somme per ogni punto di confine tra blocchi adiacenti; per l'algoritmo di debloccatura, invece, il miglioramento della qualità è in genere più marcato ed il costo computazionale è ancora inferiore poiché per ogni punto di confine tra blocchi bastano due moltiplicazioni e cinque somme; l'algoritmo adattativo, infine, dà risultati lievemente migliori (specialmente per rapporti di compressione molto elevati) con un numero di operazioni di poco superiore al precedente. Per una più completa comparazione della pesantezza computazionale dei metodi si può compilare una tabella contenente le stime del numero di operazioni necessarie per l'applicazione di ciascuno di essi nel caso di un'immagine di dimensione 256x256 e blocchi 8x8.

	Passa Basso (5x5 coefficienti)	PB solo bordi (5x5 coefficienti)	Deb. Alg.	Adap. Deb. Alg. (es. Nlun=7000)
Moltiplicazioni	1638400	793600	31744	38744
Somme	1638400	793600	79360	125104
Totale	3276800	1587200	111104	163848

5. Riferimenti bibliografici:

- Korada Ramkishor and Pravin Karandikar (Sasken Communication e Pace Soft Silicon - India)
"A Simple and Efficient Deblocking Algorithm for Low Bit-Rate Video Coding"
Proceedings of the International Symposium on Consumer Electronics (ISCE-2000), December 2000.
- Zhou Wang, Alan K. Bovik and Brian L. Evans (University of Texas - USA)
"Blind Measurement of Blocking Artifacts in Images"
Proceedings of the IEEE Int. Conf. on Image Processing (Sept. 10-13 2000), vol. III, pp.981-984.

NOTE SUL PRESENTE DOCUMENTO E SULLE MODALITA' DI UTILIZZO:

- Questo documento è il frutto del lavoro di due studenti del corso di laurea in Ingegneria Elettronica dell'Università degli Studi di Trieste e nessun docente, ricercatore o altro esperto ha partecipato alla sua stesura, dunque va letto e utilizzato tenendo conto di questo fatto.
- Andrea Coslovich non si assume nessuna responsabilità per il materiale pubblicato nella sua HomePage che è fornito "così com'è", senza alcuna garanzia implicita o esplicita. Detto materiale può essere utilizzato gratuitamente esclusivamente per scopi amatoriali o didattici mentre la proprietà intellettuale dei testi e delle immagini rimane degli autori.
- È ammessa la copia per uso personale delle opere pubblicate mentre la ripubblicazione-diffusione (parziale o integrale) in qualsiasi forma, con qualsiasi mezzo e per qualsiasi scopo potrà eventualmente avvenire solo dopo esplicita autorizzazione.