# Codifica video scalabile a bassa complessità

M. Cagnazzo, A. Caputo, G. Poggi, L. Verdoliva
Università Federico II di Napoli
Dipartimento di Ingegneria Elettronica e delle Telecomunicazioni
Via Claudio, 21 - 80125 Napoli
e-mail: [cagnazzo, caputo, poggi, verdoliv]@unina.it

**ABSTRACT**

*Nell'ambito della realizzazione di un "sistema integrato d'apprendimento" si riscontra l'esigenza di trasmettere su rete delle sequenze video, in tempo reale ed in modo efficiente, per una platea d'utenti molto eterogenea in termini di risorse disponibili (banda e potenza di calcolo). Per soddisfare tale esigenza è stato adottato un algoritmo di codifica video [2] scalabile e caratterizzato da bassissima complessità. L'algoritmo originario, basato sulla quantizzazione vettoriale gerarchica, è stato ottimizzato sotto vari aspetti, riducendo i tempi di calcolo e migliorando le prestazioni. Si è quindi realizzato un codec che, potendo operare sia in modalità unicast che multicast, garantisce ad ogni utente l'accesso in tempo reale e la massima qualità della sequenza ricevuta, in relazione alle risorse disponibili localmente.*

## 1 Introduzione

La possibilità di trasferire informazioni multimediali è una delle caratteristiche più interessanti del *networking*, sia dal punto di vista tecnico, dove si prefigura l'impegnativa sfida di fornire servizi avanzati e interattivi ad un gran numero d'utenti, sia dal punto di vista economico, dove altrettanto chiaramente si prevede lo sviluppo di un nuovo e vasto mercato. In particolare, nell'ambito della teledidattica, esistono diverse possibili applicazioni: si pensi ad esempio alle lezioni a distanza, alla videoconferenza, oppure alla possibilità di accedere da remoto, da parte di studenti o ricercatori, a strutture (laboratori, aule, ecc.) altamente complesse e costose, distribuite sul territorio nazionale.

Il problema che ci si pone è allora quello di rendere disponibile un sistema software che consenta ad un insieme eterogeneo d'utenti di trasmettere e ricevere sequenze multimediali in tempo reale ed in modo efficiente, cioè con la migliore qualità possibile compatibilmente con le risorse di banda e potenza di calcolo disponibili localmente. Per soddisfare questi requisiti, l'algoritmo di codifica da usare deve essere scalabile e a bassa complessità.

La bassa complessità è un requisito che richiede pochi commenti: è chiaro, infatti, che per funzionare in tempo reale su macchine non particolarmente potenti, l'algoritmo deve essere computazionalmente economico.

L'uso di un algoritmo scalabile (o stratificato) è un approccio comune per affrontare il problema di un insieme eterogeneo d'utenti che ricevono flussi multimediali [1], [2]. Un flusso (video) codificato si dice scalabile quando è decomponibile in *sottoflussi* dai quali è ancora possibile decodificare la sequenza video, seppure con una qualità di riproduzione inferiore. In pratica, un flusso scalabile è costituito da un flusso base, caratterizzato da parametri di qualità minimi, e da più sottoflussi aggiuntivi, che migliorano gradualmente la qualità complessiva.

Queste caratteristiche del flusso codificato possono essere sfruttate in fase di trasmissione per garantire un uso efficiente delle risorse disponibili, sia nel caso di trasmissione punto-punto (es., accesso di uno studente ad un laboratorio remoto), in cui si usa la modalità di trasmissione *unicast*, sia nel caso di trasmissione punto-multipunto (es., lezioni a distanza), in cui invece la tecnica di trasmissione è di tipo *multicast*.

Nei prossimi paragrafi saranno illustrate le tecniche usate per garantire la scalabilità e la bassa complessità dell'algoritmo di codifica; si parlerà poi di come è stato possibile migliorare le prestazioni del codec video grazie ad un insieme di tecniche innovative; infine saranno illustrati problemi e soluzioni relativi alla trasmissione del flusso codificato sulla rete.

# 2 Il codificatore video

In ogni algoritmo di codifica video la compressione del segnale d'ingresso è ottenuta sfruttando la ridondanza spaziale e la ridondanza temporale del segnale video. Con la prima espressione s'intende che le varie parti di un fotogramma sono simili tra loro. Con la seconda, ci si riferisce invece alla somiglianza tra fotogrammi consecutivi. Gli algoritmi che sfruttano questi tipi di ridondanza sono detti algoritmi di compressione spaziale e di temporale, rispettivamente.

La scalabilità viene supportata in tre forme: scalabilità in frame-rate (numero di fotogrammi al secondo), in risoluzione, in qualità (valutata tramite il rapporto segnale rumore).

## 2.1 Tecniche di compressione spaziale: la quantizzazione vettoriale gerarchica tabellare

### *2.1.1 La quantizzazione vettoriale (VQ)*

La VQ si può vedere come una generalizzazione dell'operazione di quantizzazione dal caso scalare, in cui si tratta un campione per volta, al caso vettoriale in cui più campioni d'ingresso vengono quantizzati congiuntamente. Tale operazione si può riguardare come un'applicazione di $R^k$ in $\mathbf{C}=\{\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_N\}$, un insieme discreto di elementi di $R^k$, detto *codebook*.

Nel caso del segnale video, i campioni d'ingresso sono i valori di luminanza dei fotogrammi che costituiscono la sequenza, quindi i vettori sono *blocchi di pixel*. Il codebook è costituito da vettori campione detti *codeword*. In codifica, ogni vettore d'ingresso $\mathbf{x}$ è confrontato con tutti i vettori $\mathbf{y}_i$ del codebook, per individuare quale tra questi è più simile ad esso. A questo scopo si calcola la distanza euclidea tra $\mathbf{x}$ ed ognuno degli $\mathbf{y}_i$. L'output del codificatore è l'indirizzo della codeword più vicina a $\mathbf{x}$. Tale indirizzo viene trasmesso al decodificatore, che accede al codebook e preleva la codeword ad esso corrispondente. La compressione viene ottenuta perché invece di trasmettere tutti i valori di luminanza del vettore vengono trasmessi soli i bit rappresentativi dell'indirizzo della codeword.

Descritta in questi termini, la VQ non è adatta ad un applicazione in tempo reale, visto che il calcolo della distanza di ogni vettore d'ingresso da ogni codeword può essere molto oneroso.

### *2.1.2 La VQ tabellare gerarchica*

Per ridurre la complessità, nella VQ tabellare si decide a priori come ogni possibile vettore di ingresso debba essere quantizzato, e quindi si memorizza il risultato in una tabella. Al momento della codifica non sarà necessaria nessun'operazione matematica, ma basterà accedere alla tabella per ottenere l'indirizzo che codifica il vettore. Questa tecnica, sebbene interessante, va sicuramente affinata. Infatti, se si volessero quantizzare immagini con 256 toni di grigio, usando vettori di 8 pixel, bisognerebbe costruire una tabella con $256^8$ campi (circa 16 miliardi), uno per ogni possibile vettore d'ingresso. Si può superare quest'ostacolo computazionale grazie ad un approccio gerarchico [2], [3]. Per comprenderne in funzionamento, si consideri l'esempio di figura 1. Un vettore di 8 pixel viene suddiviso in coppie, ed ognuna è codificata con la VQ tabellare, che in questo caso richiede tabelle di soli 64 kB (ogni campo corrisponde a un solo byte se si usano codebook di 256 elementi). Le uscite di queste tabelle sono raggruppate in coppie, ognuna delle quali rappresenta quindi 4 pixel dell'immagine originaria. Tali coppie possono essere ulteriormente quantizzate con tabelle sempre di 64 kB. Si possono quindi usare ulteriori stadi, ognuno dei quali raddoppia il rapporto di compressione. La complessità dell'algoritmo, al crescere del numero degli stadi, si riduce ad un accesso in tabella per ogni

campione, molto inferiore a quella della VQ convenzionale, a costo di una lieve degradazione delle prestazioni, dovuta al fatto che ogni stadio introduce rumore di quantizzazione.

### 2.2 Tecniche di compressione temporale: il Conditional Replenishment (CR)

Questa tecnica consente una buona compressione temporale (almeno per sequenze video di tipo videoconferenza, in cui il movimento è limitato), ed ha una complessità ridotta rispetto ad algoritmi di Motion Compensation. Nel CR, ogni fotogramma della sequenza è suddiviso in blocchi di pixel, detti *macroblocchi*, ognuno dei quali è confrontato con quello nella stessa posizione in una frame di riferimento (normalmente, quella precedente). Se i due macroblocchi sono abbastanza simili (la loro distanza euclidea è inferiore ad una soglia prefissata), si può evitare di trasmettere quello nuovo, e si comunica al ricevitore di recuperare le informazioni relative ad esso dal fotogramma di riferimento. Questo compito è assolto dai cosiddetti *bit di sincronizzazione* che, per ogni macroblocco, indicano l'esito del test. Se il CR fallisce, oltre ai bit di sincronizzazione, bisogna trasmettere tutti gli indirizzi che codificano il macroblocco.

### 2.3 La scalabilità in frame-rate

La scalabilità in frame rate consiste nella possibilità di ricevere solo una frame ogni N ed essere comunque in grado di decodificare correttamente la sequenza video. Bisogna però ricordare che il CR introduce dipendenza nella codifica e nella decodifica delle frame, quindi bisogna assicurare che in ogni caso l'utente sia in grado di decodificare le frame che riceve. Il problema è stato risolto usando i livelli temporali illustrati in figura 2.

Sono disponibili tre livelli temporali. Un utente può decidere se ricevere solo una frame ogni quattro, una ogni due, oppure tutte, e, corrispondentemente, riceverà solo il primo livello temporale, solo i primi due, oppure tutti. Le frame sono raggruppate quattro alla volta e quindi codificate: si codificano le frame da k+1 a k+4, poi quelle da k+5 a k+8 e così via. La codifica della frame k+4 non può avvenire usando come riferimento nessuna delle frame k+1, k+2 e k+3, perché chi riceve solo il primo livello non ne dispone. Allora si usa, come riferimento per il CR, la frame k. La frame k+2 può essere codificata dopo la k+4. Allora è possibile effettuare un CR bidirezionale[1], usando come riferimenti le frame k e k+4 (ma non k+1 e k+3). Infine, le due frame del terzo livello temporale possono essere codificate per ultime, ed entrambe con il CR bidirezionale.

### 2.4 La scalabilità in risoluzione

La scalabilità in risoluzione è ottenuta utilizzando la *codifica piramidale* (vedere la figura 3). L'obiettivo è quello di permettere all'utente di ricavare dal flusso codificato una sequenza video alla stessa risoluzione dell'originale, a risoluzione più bassa oppure a risoluzione maggiore. Dato un fotogramma a risoluzione originale, la versione a bassa risoluzione si ottiene facilmente con filtraggio e sottocampionamento. I fotogrammi a bassa risoluzione vanno a costituire il "livello base", che tutti gli utenti ricevono. L'utente che vuole una maggiore risoluzione deve ricevere anche il "livello enhancement", nel quale l'immagine a risoluzione originale è codificata con una tecnica predittiva a partire dall'immagine a bassa risoluzione. La predizione è ottenuta semplicemente interpolando l'immagine del livello base. Come in tutti gli schemi predittivi, si trasmette l'errore di predizione, cioè la differenza tra i dati predetti e quelli veri. In questo caso l'errore di predizione è costituito dai dettagli che si sono persi nella fase di filtraggio e sottocampionamento, e che ovviamente non possono essere recuperati tramite l'interpolazione. Infine, se in ricezione si dispone di abbondante potenza di calcolo è possibile effettuare un ulteriore passo d'interpolazione, ottenendo la sequenza a risoluzione più alta di quella originale.

[1] Il CR bidirezionale è una semplice estensione dell'algoritmo descritto nel paragrafo 2.2: ogni macroblocco da codificare viene confrontato con *due* macroblocchi di riferimento, tra i quali si considera solo quello più simile.

### 2.5 La scalabilità in qualità (SNR)

La scalabilità in qualità è ottenuta facendo uso di un'opportuna gerarchia di codebook con struttura ad albero (TSVQ) [4]. In questo modo si rende possibile la decodifica del flusso video anche usando un numero di bit minore di 8 per codificare gli indirizzi delle codeword.

Per capire come si raggiunge questo risultato, consideriamo come si codifica un vettore nella TSVQ. Si hanno n codebook, con cardinalità da $2^1$ a $2^n$, ognuno dei quali corrisponde ad un livello di un albero binario. I nodi dell'albero sono associati alle codeword. Il vettore d'ingresso è confrontato prima con i due vettori che costituiscono il codebook di cardinalità 2. In base al risultato si stabilisce il primo bit di codifica e si scende nell'albero al livello successivo. A questo punto l'algoritmo si ripete, confrontando il vettore d'ingresso con i due vettori "figli" della prima codeword scelta: la codifica consiste quindi in una sequenza di decisioni binarie, che consentono di accedere a codebook di cardinalità via via maggiore, raffinando gradualmente la codifica del vettore d'ingresso. In decodifica si può usare un qualsiasi prefisso dell'indirizzo della codeword: maggiore il numero di bit utilizzati, migliore (in media) la qualità della sequenza decodificata.

## 3 L'ottimizzazione del codificatore

Le tecniche descritte in precedenza sono tutte ben assestate nella letteratura scientifica, e consentono di realizzare un codec video scalabile ed a bassa complessità, che garantisce delle prestazioni accettabili, tenuto conto dei forti vincoli di progetto. Queste tecniche sono state ulteriormente migliorate, usando strategie di codifica innovative, ed in particolare ricorrendo ad elaborazioni nello spazio degli indirizzi. L'osservazione di partenza è che si dispone di un codebook *ordinato*, perché dotato di struttura ad albero, nel quale cioè le codeword con indirizzi vicini sono anche *simili* tra loro. Questo fenomeno permette diverse utili modifiche.

Anzitutto, se consideriamo due macroblocchi già codificati, possiamo stabilire se sono simili anche senza decodificarli, ma solo osservando gli indirizzi che li rappresentano. In questo modo il test per il CR è molto più veloce, perché si opera sui dati già compressi; tuttavia, lavorando sugli indirizzi si perde sensibilità sull'errore effettivamente introdotto dal CR. Gli esperimenti condotti tuttavia hanno accertato che la perdita di prestazioni è trascurabile, mentre l'aumento di velocità è sostanziale.

In secondo luogo, quando il codebook è ordinato, il codificatore VQ emette simboli correlati: infatti, per codificare una regione relativamente omogenea (come frequentemente accade nella codifica d'immagini), saranno usate codeword simili tra loro, quindi con indirizzi vicini. La correlazione degli indici può essere usata per ottenere un'ulteriore compattazione [5]. Si usa una codifica predittiva, secondo la quale l'indirizzo corrente è predetto come quello precedente sulla stessa riga o sulla stessa colonna (la scelta è affidata ad un semplice algoritmo adattativo) e l'errore di predizione è poi codificato con un codice di Huffman. Questa tecnica consente una rilevante riduzione del tasso (15 ÷ 20%) a livello base.

Vista l'importanza dell'ordinamento del codebook, si è cercato di migliorare le prestazioni del codec enfatizzando tale proprietà. Sono state seguite due strade: il riordino dei codebook esistenti e la generazione di nuovi codebook con algoritmi finalizzati all'ordinamento. Quest'ultima tecnica si è dimostrata la più efficace. I nuovi codebook sono stati generati usando l'algoritmo di Kohonen [6].

L'ottimizzazione è stata perseguita anche con una serie di modifiche meno strutturali (come ad esempio l'uso di una metrica semplificata per il test del CR), ma comunque incisive sulle prestazioni. Alcuni risultati sono riportati nei grafici di figura 4. Rispetto al codec di riferimento, descritto nella sezione 2, il tempo di elaborazione è quasi dimezzato (da 55 a 30 ms/fotogramma su un PC Pentium III a 800 MHz) mentre il PSNR aumenta di quasi un dB a tutti i bit-rate.

# 4 La trasmissione del flusso codificato

Nei precedenti paragrafi è stata descritta la struttura del codec, la cui caratteristica saliente è senza dubbio l'elevata scalabilità (nonché la possibilità di funzionare in tempo reale). Questa peculiarità viene ovviamente sfruttata nel sistema di trasmissione su rete. Sono stati implementati due prototipi che si differenziano per la modalità di trasmissione utilizzata, unicast (trasmissione uno ad uno) nel primo caso multicast (uno a molti) nel secondo.

Il flusso codificato viene diviso in diversi layer così come mostrato dalla figura 5 (si noti che in questa figura si trascura la scalabilità in qualità). Le frecce indicano in che modo si susseguono le varie operazioni di codifica. I blocchi scuri, che rappresentano i sei layer, racchiudono al loro interno i blocchi di codifica che producono il bit-stream dei vari layer. Il modo in cui questi layer vengono inoltrati verso i client destinazione è diverso a seconda del prototipo utilizzato.

## 4.1 Il prototipo unicast

Il prototipo unicast è in grado di sfruttare appieno la scalabilità offerta dal codec permettendo di variare il bit-rate. Come protocollo di trasporto si è scelto di utilizzare RTP su UDP. Sono state realizzate due applicazioni, un server ed un client, il primo si occupa di leggere le frame non codificate da disco, codificarle e trasmetterle sulla rete, mentre il client riceve queste frame, le decodifica e le visualizza in una finestra, riproducendo quindi la sequenza video, il tutto in tempo reale. I due host si accordano sui parametri della trasmissione (risoluzione, frame rate e qualità), e i layer corrispondenti vengono convogliati in un unico flusso destinato al ricevitore.

## 4.2 Il prototipo multicast

Quando si utilizza il multicast sorge un ulteriore problema, che i diversi host che ricevono hanno caratteristiche diverse tra di loro. Trasmettendo il flusso codificato su un unico gruppo multicast si va incontro al problema indicato come "scenario del minimo comune denominatore", se invece si dispone di un codec che produce un output suddiviso in layer è possibile aggirare il problema utilizzando la tecnica detta *Multiple Multicast Group* (MMG). Nel nostro lavoro si è optato per quest'ultima soluzione.

### *4.2.1 Lo scenario del minimo comune denominatore*

Utilizzando un semplice sistema di codifica con tasso variabile il trasmettitore dovrebbe scegliere i parametri di codifica che producono il flusso che si adatta alle caratteristiche dell'host dalle prestazioni peggiori . In figura 6a è mostrata una situazione di questo tipo: accanto ai link in grassetto è indicata la loro capacità, le linee tratteggiate invece rappresentano il flusso prodotto dal codificatore ed il valore accanto a esse è l'ampiezza di banda occupata da questo flusso. Come si vede tutti gli host sono costretti a ricevere un flusso a 128 kbit/s, anche se la loro rete di accesso permetterebbe di sostenere tassi superiori, situazione tipicamente indicata come scenario del minimo comune denominatore.

### *4.2.2 Multiple Multicast Group (MMG)*

Abbiamo anticipato che il codificatore è dotato di tre diversi tipi di scalabilità ed è in grado di combinarli in un flusso codificato embedded; l'idea allora consiste nel suddividere questo flusso in più sottoflussi, di cui uno è indispensabile mentre gli altri contengono le informazioni di miglioramento. Associando ciascuno di questi sottoflussi ad un diverso gruppo multicast si rendono possibili situazioni nelle quali ciascun host sceglie quali gruppi sottoscrivere, indipendentemente da quelli scelti dagli altri host, e quindi sceglie a quale tasso ricevere. Lo schema ottenuto prende il nome di Multiple Multicast Group (MMG), o anche di Receiver-driven Layered Multicast (RLM) [1] in quanto la scelta è appunto effettuata dal ricevitore e non più dal trasmettitore che in questo caso trasmetterà sempre alla massima qualità possibile senza

nemmeno sapere cosa hanno scelto i ricevitori. In figura 6b è mostrata una situazione in cui si applica la tecnica MMG. Ogni host è libero di decidere la qualità del flusso ricevuto in modo indipendente dagli altri. In questo modo si riesce a soddisfare le richieste di host dalle caratteristiche eterogenee.

Non sono state implementate particolari tecniche per la protezione degli errori, anche perché la stessa tecnica del CR mostra una certa "robustezza" nei confronti delle perdite di pacchetti.

Entrambi i prototipi sono stati dotati di un'interfaccia grafica (Fig.7) che permette di modificare in modo semplice ed intuitivo i parametri di codifica e allo stesso tempo, nell'ottica di un uso anche didattico, permette di valutare in tempo reale i risultati delle scelte effettuate.

## 5 Conclusioni e sviluppi futuri

Il codec di Chaddha e Gupta, basato sulla VQ tabellare gerarchica, permette di soddisfare molti dei requisiti della trasmissione video su reti eterogenee. In questo lavoro si sono proposte e analizzate alcune modifiche rispetto all'algoritmo originale [2] che hanno permesso di ridurre i tempi di calcolo e migliorare le prestazioni di codifica.

Esistono certamente altri margini di miglioramento: ad esempio è possibile effettuare per via tabellare anche campionamento ed interpolazione che, allo stato attuale, restano le operazioni più complesse. Inoltre, si pensa di sperimentare in futuro l'uso della decomposizione wavelet al posto della piramide gaussiana, in quanto non ridondante e quindi, presumibilmente, più efficiente.

## Bibliografia


[1] S. McCanne, M. Vetterli, and V. Jacobson, "Low-Complexity Video Coding for Receiver-Driven Layered Multicast", *IEEE Journal Select. Areas Commun.*, vol. 15 pp.983-1000, August 1997.

[2] N. Chaddha, and A. Gupta, "A Frame-work for Live Multicast of Video Streams over the Internet", *International Conference on Image Processing*, 1996.

[3] N. Chaddha, M. Vishvanath, and P. A. Chou, "Hierarchical Vector Quantization of Perceptually Weighted Block Transforms", *Proc. of data Compression Conference*, March 1995

[4] N. Chaddha, P. A. Chou, and R. M. Gray, "Constrained and recursive Hierarchical Table-Lookup Vector Quantization", *Proc. of data Compression Conference*, April 1996

[5] G. Poggi, "Address-Predictive Vector Quantization of Images by Topology-Preserving Codebook Ordering", *Euro. Trans. Telecommun.*, vol. 4, pp. 423-434, July-August 1993.

[6] T. Kohonen. *Self-Organization and Associative Memory*, 2nd ed., Springer Verlaf, 1988.

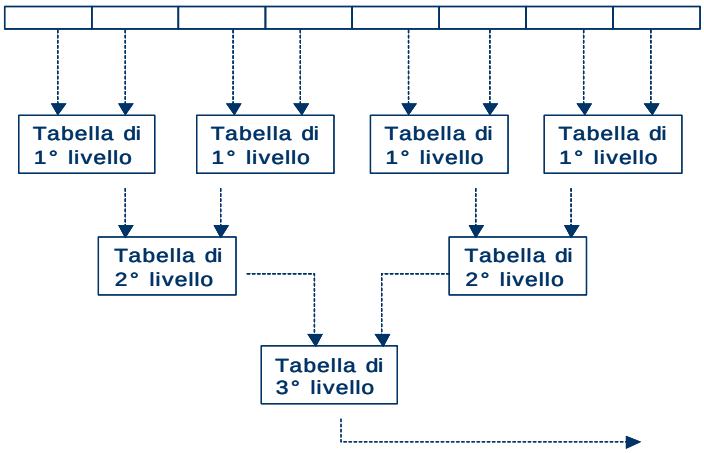


**Figura 1. Schema della VQ gerarchica tabellare**

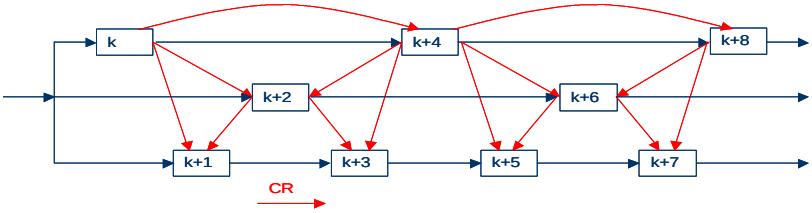


**Figura 2. Livelli temporali e scalabilità in frame rate.**

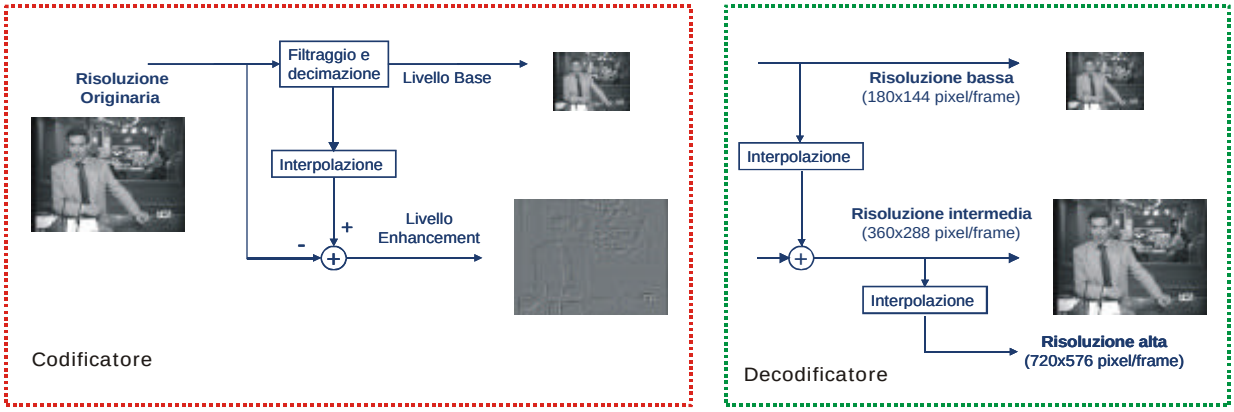


**Figura 3. Codifica piramidale**

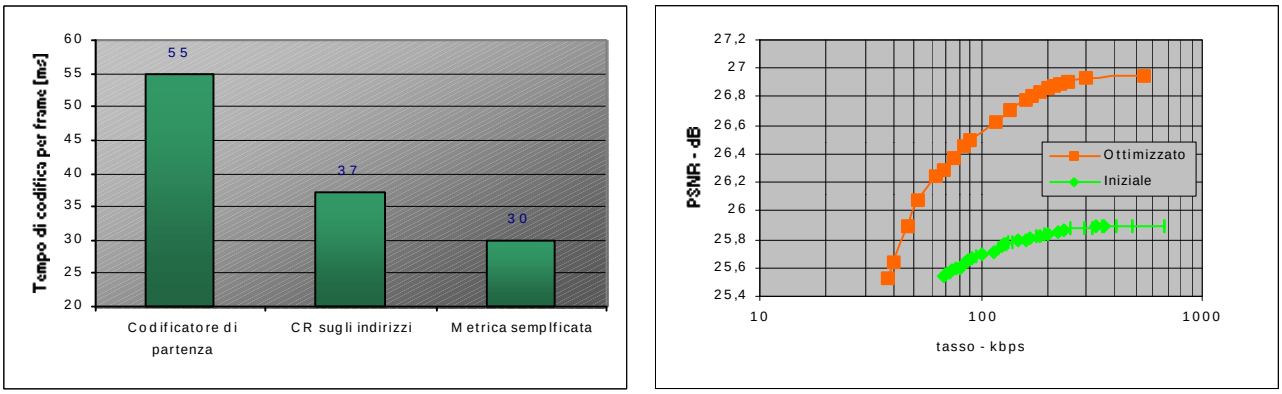


**Figura 4. Prestazioni del codec ottimizzato**

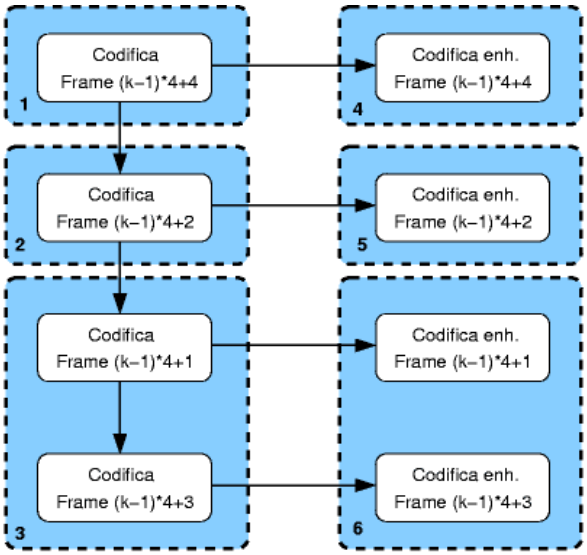


**Figura 5: La separazione in layer.**

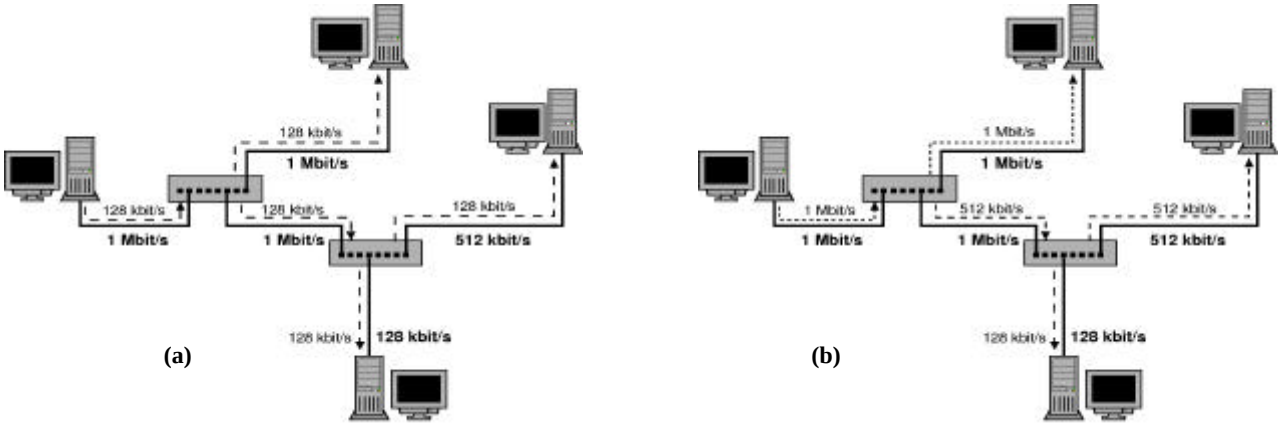


**Figura 6. Caso Multicast. (a) Scenario del minimo comun denominatore; (b) scenario MMG**

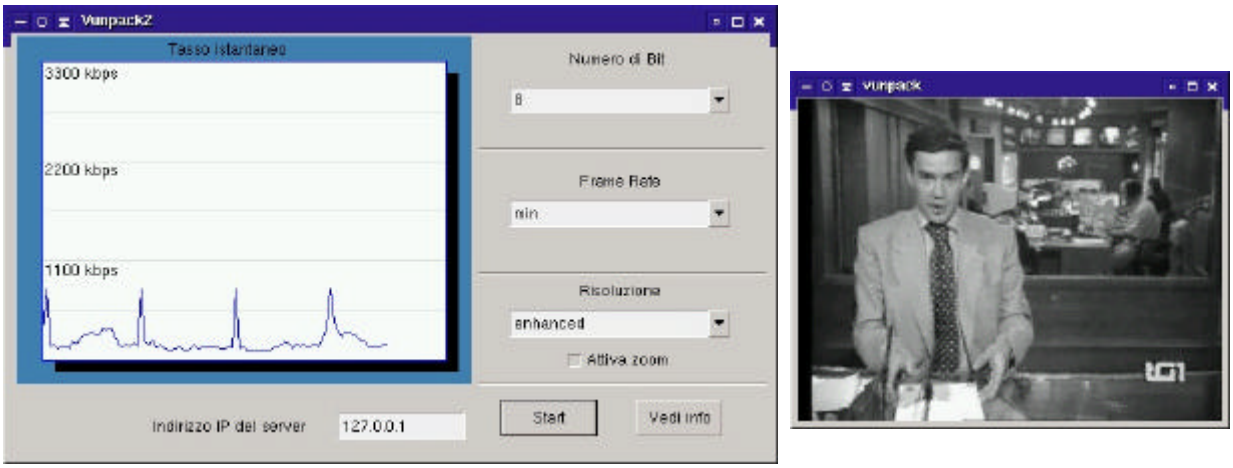


**Figura 7: L'interfaccia grafica**