Algoritmo di compressione 2d

Studio sulla variazione del coefficiente di compressione al variare della soglia alta e bassa.

Filippo Costa

Alice pag 3.
Algoritmo di compressione 2d pag 4.
Prove effettuate in laboratorio pag 6.
Visualizzazione dei dati in modalità grafica pag 8.

ALICE

ALICE (A Large Ion Collider Experiment) è un esperimento svolto al LHC (Large Hadron Collider) ottimizzato per lo studio di collisioni tra ioni pesanti, con un'energia nel centro di massa pari a 5.5 TeV per nucleone. L'obiettivo di questo esperimento è di studiare nei dettagli il comportamento della materia nucleare ad alte densità e temperature.
I detector di ALICE consistono essenzialmente in due parti principali: la parte centrale è formata da detector devoti allo studio dei segnali adronici e dei dielettroni, e più avanti lo spettrometro dei muoni, devoto allo studio del comportamento dei quark ad alte densità della materia.
Noi ci occuperemo principlamente dell'ITS (Inner Track System).
L'ITS è composto da 6 strati cilindrici (chiamati layer).
Ci occuperemo prevalentemente dei layer centrali (3 e 4).
I rivelatori a questo livello sono camere a deriva di silicio (SDD Silicon Drift Detector).
Un po' di numeri.
Ogni modulo SDD è diviso in due semirivelatori, ognuno di questi contiene 256 anodi alla distanza di 300 micrometri.
Quindi ogni modulo SDD contiene 256x2 canali di lettura, per 260 rivelatori contenuti nei layer 3 e 4 otteniamo un totale di 133k canali di lettura.
L'ammontare di informazioni generata da un SDD è molto grande: ogni mezzo detector ha 256 anodi e per ogni anodo il dato viene campionato per 256 volte.
I dati passano attraverso l'elettronica di front end ed arrivano a Carlos che li deve comprimere e spedire.
Tralasciamo i dettagli di come i dati vengono digitalizzati e successivamente spediti a Carlos.
Il semidetector visto da Carlos è una matrice 256x256 dove ogni cella contiene 8 bit di informazioni, per un totale di 256x256x8bit=64Kbyte.
Il volume totale di informazioni prodotto da tutti gli SDD ammonta circa a 32 Megabytes per evento, mentre lo spazio riservato sul disco per contenere le informazioni è di 1,5MegaBytes, questo ci obbliga ad utilizzare un algoritmo di compressione (con un coefficiente di compressione pari a 22), che ci permetta di comprimere i dati con la minor perdita di informazioni possibile.
L'algoritmo di compressione che è stato implementato su CARLOS è l'algoritmo 2d.

Algoritmo di compressione 2d

SDD rappresenta una delle 3 tecnologie in silicone adottate nell'ITS di ALICE.
Occupa i due layer centrali, 3 e 4, dove la densità delle particelle non è sufficientemente alta per giustificare l'uso di pixel in silicone e non è sufficientemente bassa per favorire la scelta di silicon strips.
SDD ha un'area attiva molto grande (7,25x7,53 cm^2) e due righe da 256 anodi una opposta all'altra.
La nuvola di carica generata da una particella ionizzata attraverso il detector si sposta al centro del detector verso la riga di anodi più vicina sotto l'azione di un campo elettrico uniforme.
I segnali generati da ogni particella devono essere analizzati per poter misurare le due coordinate del centro della nuvola di carica con una precisione di circa 30μm e l'ammontare totale della carica con una risoluzione migliore del 10%.
Conseguentemente il segnale anodico deve essere campionato alla frequenza di 40 Mhz e digitalizzato in parole da 10 bit.
Chiamiamo *cluster* il set digitalizzato corrispondente alla nuvola di carica.
Ogni SDD produce qualche cosa come 256x10bit di informazione per evento-->circa 40,6 Megabytes.
Prima di trasmettere i dati al *DAQ* una sostanziale compressione dei dati deve essere eseguita per soddisfare le costrizioni imposte dalla memoria di massa disponibile (1,5 Megabytes).
Allora è richiesto circa un coefficiente di compressione pari a 22 (in realta l'informazione prodotta da ogni SDD quando arriva a CARLOS non è di 256x10bit, ma bensì 256x8bit-->circa 32 Megabytes per evento, ecco spiegato il coefficiente 22).
A causa della presenza di rumore un algoritmo di compressione con una sola soglia non è sufficientemente efficace. Infatti per ottenere un coefficiente di compressione pari a 22 sono richiesti valori molto alti della soglia (circa 3 RMS), questo comporta una perdita notevole di informazioni utili.
Ecco la necessità di un nuovo algoritmo di compressione (1d-->2d) che utilizzi non una soglia, ma bensì 2.
Deve soddisfare tre proprietà importanti:

1. Deve essere mantenuta la massima precisione dopo la conversione ADC così che tutte le informazioni del campione siano disponibili nel cluster di analisi.
2. I valori dei campioni nelle vicinanze del cluster devono essere disponibili per un'accurata misura delle caratteristiche del cluster, per una buona monitorizzazione e comprensione delle caratteristiche del background.
3. La natura statistica dei campioni soppressi deve essere disponibile per monitorare il livello di rumore degli anodi e per ottenere il loro valore base, che deve essere sottratto dai campioni del cluster per ottenere una misura corretta della relativa carica.

L'algoritmo 2d, vede il detector SDD come due matrici da 256x256 elementi e ogni elemento della matrice rappresenta una quantità codificata in 8 bit.
In questo array i segnali delle particelle sono rappresentati da cluster di campioni adiacenti, i cui valori stanno oltre il background.
I poteziali cluster sono cercati esaminando riga per riga tutti i quintetti di campioni disposti a croce.
Quindi abbiamo 3 campioni adiacenti su una riga e 3 campioni adiacenti su una colonna, si intersecano formando una croce.
Il campione centrale (che indicheremo con C)appartiene al cluster se soddisfa la soglia Tl, e se il quintetto contiene almeno un campione che soddisfi la soglia Th e almeno due che soddisfino la soglia Tl. La matrice viene letta pixel per pixel e ogni volta si controlla se il pixel letto può essere un potenziale centro di una croce (NSWEC).
I singoli pixel (chiamati “single high background peaks) sono soppressi a 0 se non hanno nessuno nelle vicinanze che soddisfi almeno la soglia bassa, così da non sovraccaricare l'acquisizione dei dati e permettere una efficiente sopressione degli 0.

Lo sviluppo di questo algoritmo a doppia soglia e una soppressione degli 0 a 2 dimensioni:

- mantiene massima precisione con un numero minimo di bit per i cluster campionati
- considera la vicinanza sia lungo l'asse degli anodi che lungo l'asse del tempo
- colleziona dati a sufficienza per caratterizzare il background soppresso

Le prove in laboratorio hanno dimostrato le ottime capacità di compressione e la bassa perdita di informazioni utili.

Una descrizione dettagliata del funzionamento dell'algoritmo si trova su un altro documento in cui descrivo il funzionamento del programma che simula il compressore montato su CARLOS.

Prove effettuate in laboratorio

Mi è stato dato un programma in C++ che simulava il comportamento dell'algoritmo 2d implementato sulla scheda CARLOS.

Il programma in questione prende in input un file contenente degli eventi dell'esperimento ALICE li comprime e mette in output una serie di informazioni.....quali la parola compressa, i numeri di bit da considerare buoni di quella parola.

Mi sono stati dati 2 file, uno contenente 499 eventi, l'altro 500.

I vari test sono stati svolti in questo modo:

- ho studiato il funzionamento del programma.
- Ho esaminato la struttura dei file contenenti gli eventi. I dati in questi file sono disposti in righe da 32 interi per riga, quindi 8 righe di numeri formano una riga di anodi. Ogni evento ha un titolo che lo identifica dal successivo
- ho dato in input il file al programma utilizzando valori della soglia alta e bassa sempre diversi per vedere come variava il coefficiente di compressione
- con l'aiuto di alcuni programmi scritti da me ho analizzato l'output del programma e ho rappresentato i risultati in un foglio elettronico, così da avere un riscontro visivo e immediato.

L'analisi del programma che simula l'algoritmo e tutti i programmi che ho utilizzato in questa esperienza è contenuta i un altro documento che sto finendo di scrivere.

Ecco i primi risultati ottenuti dalle mie analisi.

I primi test sono stati eseguiti sul file "**monocluster.raw**".

Questa tabella rappresenta il valore del coefficiente di compressione al variare della soglia alta e bassa.

Th	Tl=0	Tl=4	Tl=5	Tl=6	Tl=7	Tl=8	Tl=9	Tl=10	Tl=11	Tl=12	Tl=13	Tl=14	Tl=17	Tl=20	Tl=23	Tl=26	Tl=29
18	1,95	1,95	1,2	1,2	1,24	1,27	1,29	1,32	1,35	1,39	1,44	1,51	1,86				
19	1,22	1,22	1,24	1,25	1,27	1,3	1,32	1,35	1,38	1,42	1,47	1,54					
20	1,27	1,27	1,29	1,3	1,33	1,35	1,37	1,4	1,44	1,49	1,54	1,61	1,98	3			
21	1,4	1,4	1,4	1,4	1,44	1,5	1,5	1,53	1,57	1,6	1,67	1,75					
22	1,58	1,6	1,6	1,6	1,6	1,66	1,7	1,72	1,77	1,83	1,9	2	2,41	3,37			
23	1,92	1,9	1,93	1,95	1,97	2	2	2,1	2,14	2,2	2,3	2,4			7		
24	2,4	2,4	2,44	2,46	2,48	2,5	2,56	2,6	2,7	2,76	2,87	3	3,6	4,76	7,6		
25	3,2	3,17	3,2	3,2	3,25	3,3	3,34	3,4	3,5	3,6	3,73	3,9					
26	4,56	4,56	4,6	4,6	4,67	4,7	4,8	4,88	5	5,14	5,33	5,58	6,67	8,42	11,4	22	
27	6,76	6,76	6,8	6,85	7	7	7,1	7,22	7,4	7,6	7,86	8,21					
28	10,47	10,47	10,5	10,6	10,7	10,8	11	11,15	11,38	11,7	12,1	12,59	14,78	17,96	21,93	29,73	
29	16,47	16,47	16,6	16,7	16,8	17	17,2	17,48	17,8	18,24	18,82	19,53					63,6
30	25,66	25,66	25,8	26	26,2	26,4	26,7	27	27,54	28,1	28,87	29,79	33,6	38,84	44	50,15	67,6

N.B. Per il momento mi sono fermato ad un valore massimo di Th =30, poiché ho notato che per Th=30 e Tl=0 si raggiunge un valore di compressione pari a 25,66 che è maggiore del coefficente richiesto (22).

Esprimendo il valore della soglia Ti (con i=H,L) come

$$Ti=u+fs$$

u=media dei dati contenuti negli eventi

s=devizione standard

f=fattore moltiplicativo
Otteniamo una stima di quanta informazione utile viene buttata via durante la compressione.
Con il compressore 1d ottenevamo buoni valori di compressione solo per f=3, ma era un valore troppo alto, si buttavano via troppe informazioni utili, quindi considereremo buoni tutti i valori di f sotto a 3.
Con la porssima tabella vado a rappresentare il variare di f con la soglia Ti.

Ti	Fat. Moltiplicativo
18	0,17
19	0,37
20	0,57
21	0,78
22	0,98
23	1,2
24	1,38
25	1,58
26	1,8
27	2
28	2,2
29	2,4
30	2,6

Vediamo che i valori di soglia rimangono tutti sotto al 2,6.

Ho ripetuto le stesse prove anche su un altro file ("**multicluser.raw**") e questi sono stati i risultati.

Th	Tl=8	Tl=11	Tl=14	Tl=17	Tl=20	Tl=23	Tl=26	Tl=29
18	1,27	1,36	1,52	1,88				
19								
20	1,35	1,45	1,62	2	2,93			
21								
22	1,62	1,73	1,95	2,36	3,25			
23						6,3		
24	2,31	2,46	2,77	3,33	4,32	6,7		
25								
26	3,87	4,1	4,6	5,48	6,9	9,26	16,7	
27								
28	7,42	7,84	8,7	10,23	12,5	15,46	21	
29								42,16
30	15,2	15,97	17,44	19,95	23,65	27,59	32,4	44,62

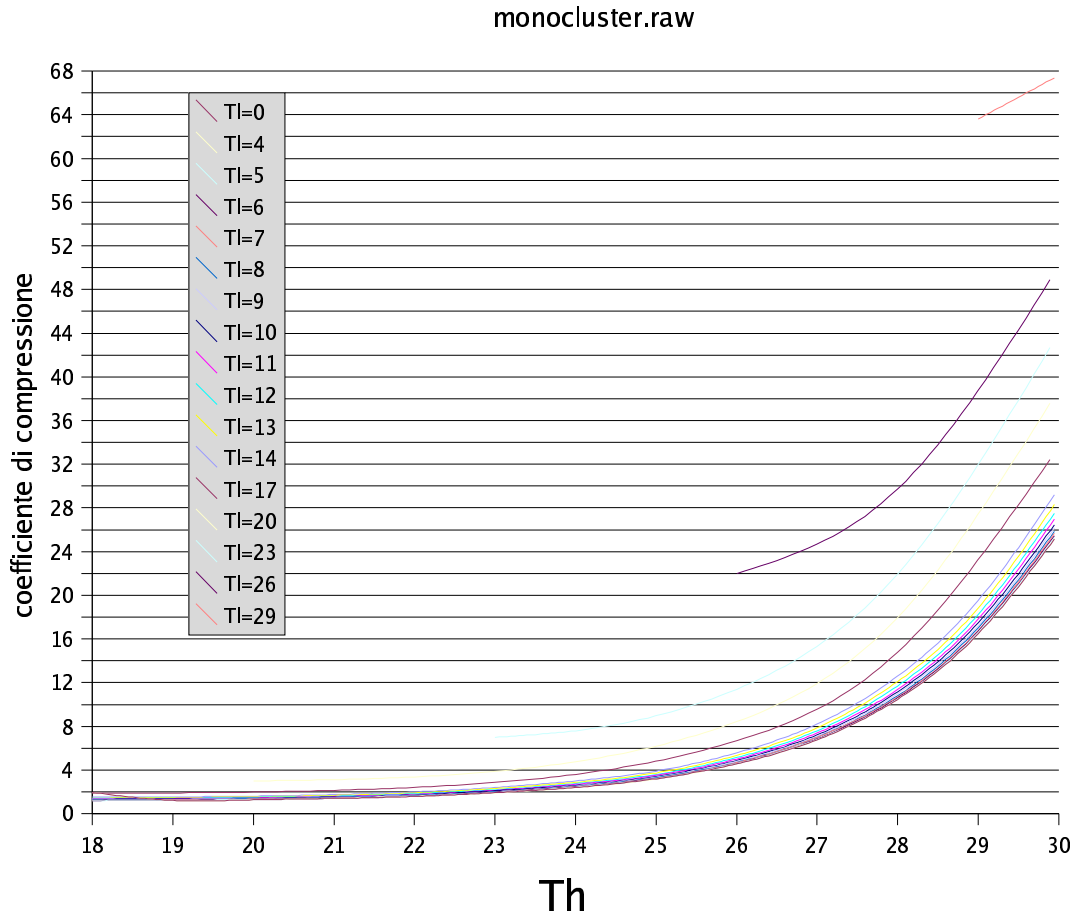
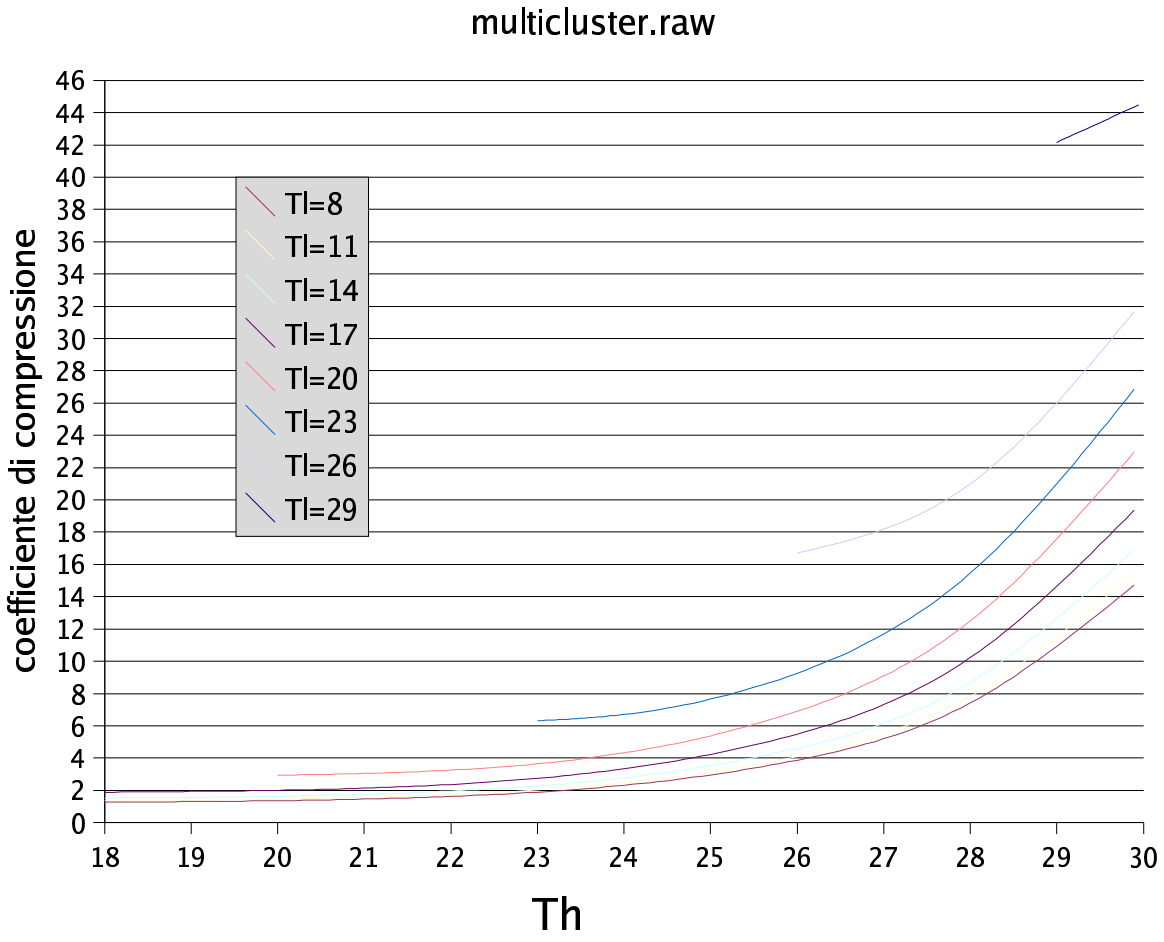
Possiamo notare che per fino a Tl=20, non si raggiunge un coefficiente di compressione sufficientemente elevato.

Ti	Fat. Moltiplicativo
18	0,14
19	0,31
20	0,48
21	0,65
22	0,83
23	1
24	1,17
25	1,34
26	1,52
27	1,7
28	1,86
29	2,03
30	2,2

Anche in questo caso rimaniamo sempre sotto a 3.

Visualizzazione dei dati in modalità grafica

Th	Tl=0	Tl=4	Tl=5	Tl=6	Tl=7	Tl=8	Tl=9	Tl=10	Tl=11	Tl=12	Tl=13	Tl=14	Tl=17	Tl=20	Tl=23	Tl=26	Tl=29
18	1,95	1,95	1,2	1,2	1,24	1,27	1,29	1,32	1,35	1,39	1,44	1,51	1,86				
19	1,22	1,22	1,24	1,25	1,27	1,3	1,32	1,35	1,38	1,42	1,47	1,54					
20	1,27	1,27	1,29	1,3	1,33	1,35	1,37	1,4	1,44	1,49	1,54	1,61	1,98	3			
21	1,4	1,4	1,4	1,4	1,44	1,5	1,5	1,53	1,57	1,6	1,67	1,75					
22	1,58	1,6	1,6	1,6	1,6	1,66	1,7	1,72	1,77	1,83	1,9	2	2,41	3,37			
23	1,92	1,9	1,93	1,95	1,97	2	2	2,1	2,14	2,2	2,3	2,4			7		
24	2,4	2,4	2,44	2,46	2,48	2,5	2,56	2,6	2,7	2,76	2,87	3	3,6	4,76	7,6		
25	3,2	3,17	3,2	3,2	3,25	3,3	3,34	3,4	3,5	3,6	3,73	3,9					
26	4,56	4,56	4,6	4,6	4,67	4,7	4,8	4,88	5	5,14	5,33	5,58	6,67	8,42	11,4	22	
27	6,76	6,76	6,8	6,85	7	7	7,1	7,22	7,4	7,6	7,86	8,21					
28	10,47	10,47	10,5	10,6	10,7	10,8	11	11,15	11,38	11,7	12,1	12,59	14,78	17,96	21,93	29,73	
29	16,47	16,47	16,6	16,7	16,8	17	17,2	17,48	17,8	18,24	18,82	19,53					63,6
30	25,66	25,66	25,8	26	26,2	26,4	26,7	27	27,54	28,1	28,87	29,79	33,6	38,84	44	50,15	67,6



Th	Tl=8	Tl=11	Tl=14	Tl=17	Tl=20	Tl=23	Tl=26	Tl=29
18	1,27	1,36	1,52	1,88				
19								
20	1,35	1,45	1,62	2	2,93			
21								
22	1,62	1,73	1,95	2,36	3,25			
23						6,3		
24	2,31	2,46	2,77	3,33	4,32	6,7		
25								
26	3,87	4,1	4,6	5,48	6,9	9,26	16,7	
27								
28	7,42	7,84	8,7	10,23	12,5	15,46	21	
29								42,16
30	15,2	15,97	17,44	19,95	23,65	27,59	32,4	44,62