

SCIENZA E FANTASCIENZA

Il dato come indizio

Firenze, 7-8 maggio 2025

SEZIONE TESINE TRIENNIO

Menzione d'onore

Messaggeri del cosmo

Studenti

Patron Giacomo - Musci Matteo - Miriam Aliberti

Conti Mattia - Bisello Giulia

Classi 4A – 4C - 4D – 4E - 4G

Istituto di Istruzione Superiore

Liceo scientifico Galileo Galilei - Selvazzano Dentro (PD)

Docente Coordinatore

Rocci Alessio

La scelta dell'attività sperimentale, la rilevazione dei muoni cosmici, è supportata da analisi critica del problema e da attenta valutazione delle risorse disponibili. Ben condotte la preparazione del set sperimentale e l'elaborazione dei dati. Lodevole la decisione di applicare due metodi di raccolta e trattamento dati: uno utilizzando un apposito software, l'altro scrivendo ex novo il codice con MatLab. Il confronto tra i due procedimenti si è rivelato molto educativo e ha arricchito la fase interpretativa dei risultati sperimentali.

Messaggeri del cosmo

INDICE:

Introduzione	1
L'apparato sperimentale	2
Dati sperimentali	3
Theremino	3
MatLab	5
Problema di combinatoria	8
Analisi dei dati	9
Scrematura dati	9
Modello finale	10
Conclusioni	12
Bibliografia e sitografia	12
Appendice codice	12

Introduzione

La decisione dell'esperimento è stata una fase relativamente lunga, a causa dell'indecisione generale che la varietà delle proposte, presentate da ognuno dei componenti del gruppo, aveva generato.

Alla fine è stato deciso indagare la presenza di segnali extraterrestri sulla Terra, concentrandosi sulla rilevazione dei muoni. Si tratta di una sorta di messaggeri che portano informazioni da fenomeni che accadono dentro e fuori il nostro sistema solare. Questa caratteristica dei fratelli pesanti degli elettroni ci ha spinto a indagare il loro comportamento. Inizialmente avevamo l'obiettivo di capire se i muoni fossero in grado di viaggiare più veloci della luce, come Superman o Flash. Il nostro nostro docente referente infatti ci aveva informati del loro utilizzo nel verificare i fenomeni della relatività speciale di Einstein. Abbiamo così intrapreso il nostro viaggio alla ricerca dei segnali dal cosmo, cercando, come dice il nostro docente referente, **di arrivare dove nessun altro studente è mai giunto prima (cfr. Star Trek)** seguendo le tracce dei nostri dati sperimentali.

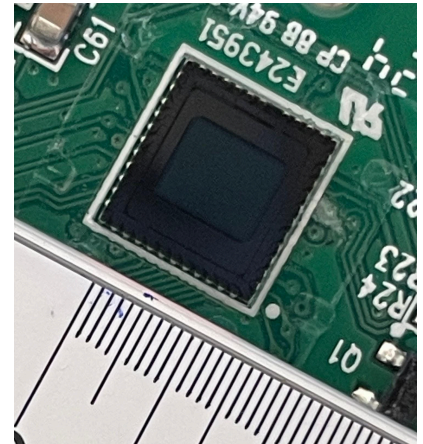
Una volta deciso l'argomento ci si sono presentati due esperimenti fattibili a scuola: uno utilizzando le webcam e l'altro con i tubi fluorescenti e abbiamo cominciato a indagare entrambi. Essi si basano sul fatto che quando particelle ad alta energia attraversano un mezzo, in un caso il CMOS e nell'altro il neon, interagiscono con questi e nel primo caso ionizza il sensore mentre nel secondo caso il neon. L'ultima opzione si è rivelata molto complicata in quanto prima di tutto gli articoli a disposizione sono pochissimi e in tutto siamo riusciti a trovare solo due articoli che spiegassero (anche se in modo molto vago) come costruirne uno [1] e [2]. La cosa che necessita una minuziosa preparazione, però, è che serve una tensione di 1kV in fili scoperti, quindi, sebbene disponiamo a scuola di un generatore da 6kV, il possibile utilizzo dell'alta tensione ci ha costretto a scrivere una relazione introduttiva per l'RSPP della scuola affinché valutasse le opzioni di rischio. Questo ha comportato un grande impiego di tempo e quindi nei primi incontri di Gennaio abbiamo rimandato questa attività a dopo la consegna di questa tesina, per poterci concentrare su un'esperienza, quale quella con le webcam di cui parleremo, più sicura. Il nostro docente referente, infatti, ci ha permesso di utilizzare le webcam oggi inutilizzate che la scuola aveva acquistato al tempo del Covid-19.

L'apparato sperimentale

Le Webcam utilizzate sono le C270 della Logitech. All'interno della webcam è presente il sensore CMOS (Complementary Metal Oxide Semiconductor), elemento sensibile alla luce e ad altre particelle ionizzanti, come i muoni [3], che sarà quindi il nostro rivelatore.

Per quanto riguarda la misura della superficie del CMOS, abbiamo provato a verificare se quelle del sito corrispondessero con quelle reali delle nostre webcam e abbiamo constatato che varia in ognuna di esse.

Per misurare la grandezza dei sensori delle webcam abbiamo fatto un confronto con una scala graduata con sensibilità 1/25cm facendo una foto tenendo l'obiettivo del telefono in corrispondenza del margine del righello e centrato con il sensore, in modo da evitare il più possibile la parallasse. Per analizzare queste foto abbiamo usato il codice presente nella guida di MatLab [4] e prima di tutto abbiamo fatto misurare da una stessa persona per 5 volte 2 millimetri sulla scala graduata per ottenere una misura di riferimento, e poi abbiamo misurato il sensore stesso. Abbiamo preso come errore assoluto della misura la semidispersione delle misure di calibrazione.



Dati del sensore CMOS:

Dimensione sensore: $(0,369 \pm 0,002) \times (0,309 \pm 0,002) \text{ cm}$

Superficie sensore: $(0,114 \pm 0,001) \text{ cm}^2$

Abbiamo inoltre oscurato il CMOS per ridurre il più possibile che altri eventi esterni interferissero con le nostre misurazioni tramite uno strato di carta da forno e uno di carta stagnola. Infatti usando il nastro adesivo, come molti fanno, ci siamo accorti che a causa del calore che le schede producono, la colla si scioglie e il nastro si stacca causando un'alterazione alle acquisizioni. Per quanto riguarda le acquisizioni in biblioteca di cui parleremo dopo, abbiamo inoltre posizionato la webcam in una scatola di un telefono con un buco per far passare i cavi in modo da ottenere un oscuramento ancora più efficiente.

All'inizio non sapevamo quanto sarebbero dovute durare ogni acquisizione per ogni angolo quindi siamo partiti con 2 minuti ma, non ottenendo dati soddisfacenti siamo passati a 5 minuti, poi 15 e infine 1 ora. Oltre un'ora per angolazione non abbiamo registrato con matlab per questioni di tempistica di esecuzione del codice. Al contrario, theremino non ha bisogno di un tempo di compilazione, quindi ci siamo potuti spingere oltre registrando dati per giorni interi. Infatti abbiamo posizionato un computer nella biblioteca della scuola sempre collegato alla corrente e ogni giorno andavamo a raccogliere i dati. Quindi ogni dato del flusso di theremino è stato registrato su un intervallo di 24 ore mentre ogni dato del flusso di matlab è stato registrati in un'ora.

La soglia di sensibilità è collegata con il numero minimo di fotoni con cui la telecamera può distinguere le informazioni utili nell'immagine dal rumore. Il rumore è il segnale generato da ogni pixel anche quando il sensore è oscurato. Ogni telecamera ha un suo rumore di fondo [5]. Abbiamo constatato che il CMOS della telecamera utilizzata da noi ha una soglia di rumore pari a 6: per non rivelare rumore abbiamo dunque impostato la soglia almeno a 7.

Abbiamo inoltre, per effettuare le misurazioni con diverse angolazioni, posizionato le webcam su un piano inclinato e in modo che fossero il più



possibile parallele ad esso, grazie ad un supporto in polistirolo all'interno di una scatola.

Nel frattempo, utilizzando un sensore Geiger-Müller della Vernier, abbiamo provato a misurare la radiazione di fondo ottenendo come valore una stima che abbiamo deciso di non considerare in quanto poco precisa dal momento che la misura stessa del sensore era una stima perché non abbiamo potuto separare il sensore dal suo supporto.

Dati sperimentali

Per raccogliere i dati abbiamo percorso due strade, una usando il programma Theremino e l'altra con MatLab cercando di creare da zero un nostro programma.

Per calcolare l'errore assoluto nel conteggio degli eventi abbiamo usato la distribuzione di Poisson o legge degli eventi rari, una distribuzione di probabilità discreta che esprime le probabilità per il numero di eventi, in media pari a λ in un dato intervallo di tempo [6].

La distribuzione è data da:

$$P_{\lambda}(n) = \frac{\lambda^n}{n!} e^{-\lambda}, \forall n \in \mathbb{N}$$

Dove:

- λ è il numero medio di eventi per intervallo di tempo
- n è il numero di eventi per intervallo di tempo

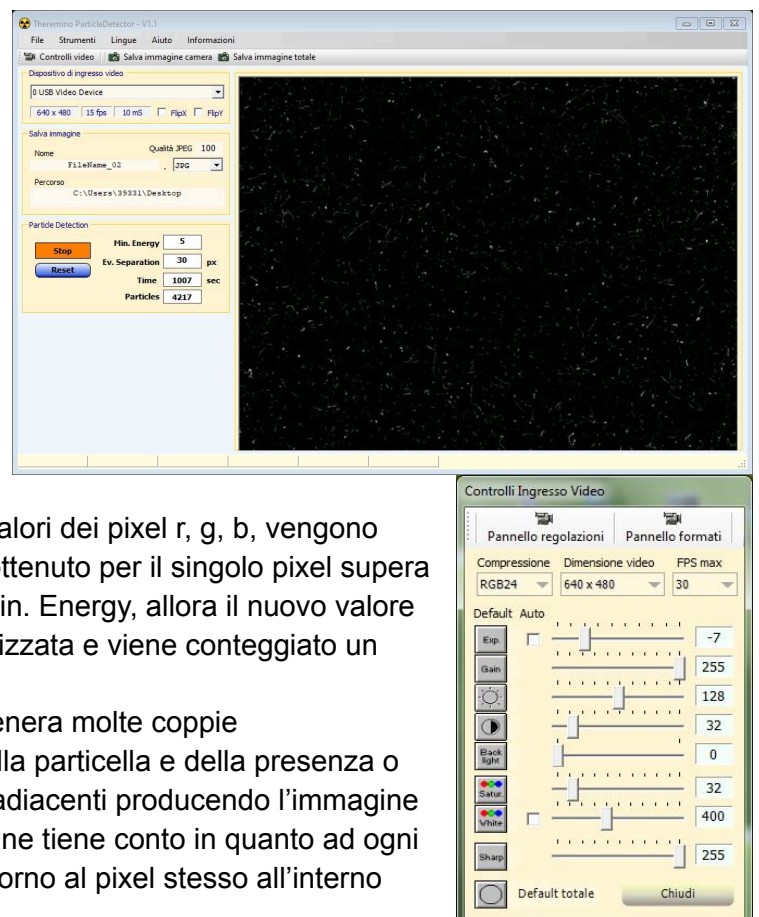
e si cerca la probabilità di n .

Si può dimostrare che per questa distribuzione la varianza è pari allo stesso valore medio λ . Nel caso degli eventi rari come il decadimento radioattivo o l'interazione dei muoni con il CMOS, dunque, l'errore, ovvero la radice della varianza, è pari alla radice del conteggio degli eventi, che rappresenta il valor medio della variabile aleatoria (variabile che assume valori diversi in dipendenza da un fenomeno aleatorio).

Theremino

Il programma Theremino Particle detector [7] permette dispositivi video, come una webcam, per creare una immagine come se fosse ottenuta con una fotocamera avente l'otturatore sempre aperto, l'applicazione infatti effettua una sovrapposizione delle immagini acquisite per creare un specie di "lunga esposizione". In questo modo le particelle quando eccitano il sensore delle webcam non scompaiono ma rimangono impresse sullo schermo. Nel programma è anche incluso un algoritmo che permette di contare gli eventi: i frame acquisiti dalle webcam vengono scansionati pixel per pixel, durante la scansione i valori dei pixel r, g, b, vengono convertiti in scala di grigi. Se il valore del segnale ottenuto per il singolo pixel supera la soglia configurata dall'utente, con il parametro Min. Energy, allora il nuovo valore per quel pixel viene inserito nella immagine memorizzata e viene conteggiato un nuovo evento.

Il passaggio di una particella attraverso il CMOS genera molte coppie elettrone-lacuna che, a seconda della traiettoria della particella e della presenza o meno di elettroni secondari, eviteranno molti pixel adiacenti producendo l'immagine di una traccia. L'algoritmo di conteggio degli eventi ne tiene conto in quanto ad ogni pixel sopra la soglia di energia assegna un'area attorno al pixel stesso all'interno



della quale eventuali altri pixel attivati non innescano l'incremento del contatore degli eventi. Questo permette di evitare conteggi duplicati nel caso di tracce che si estendono su diversi pixel. Questa area è configurabile dall'utente attraverso il parametro Ev. Separation, che corrisponde alla distanza minima che due pixel devono avere per essere conteggiati come due eventi distinti.

Abbiamo quindi due parametri che possono essere modificati durante ogni acquisizione, la Min.Energy che abbiamo variato secondo le nostre esigenze e la Ev.Separation che abbiamo sempre mantenuto a 30, come raccomandato nella documentazione del programma. Sempre nei documenti allegati a Theremino viene raccomandato di impostare le webcam in modo particolare per ottenere migliori acquisizioni, e così abbiamo fatto per ogni acquisizione.

Usando Theremino abbiamo ottenuto i seguenti risultati:

Gradi orizzontali	soglia	tempo (s)	eventi	errore eventi	flusso (CPM/cm ²)	errore flusso	ϕ (CPS/cm ²)
0	10	64104	85	9,22	0,70	0,08	0,70±0,08
0	10	174726	252	15,87	0,76	0,06	0,76±0,06
0	9	14701	24	4,90	0,86	0,19	0,9±0,2
0	9	75640	84	9,17	0,58	0,07	0,58±0,07
0	7	87774	236	15,36	1,42	0,11	1,4±0,1
0	8	81904	207	14,39	1,33	0,11	1,3±0,1
40	10	167924	407	20,17	1,28	0,08	1,28±0,08
40	10	78187	105	10,25	0,71	0,08	0,71±0,08
40	10	80100	124	11,14	0,81	0,08	0,81±0,08
180	14	268207	454	21,31	0,89	0,05	0,89±0,05
180	10	96710	165	12,85	0,90	0,08	0,90±0,08
180	10	69229	139	11,79	1,06	0,10	1,1±0,1
180	10	103590	148	12,29	0,76	0,07	0,76±0,07
180	7	19538	52	7,21	1,40	0,21	1,4±0,2

MatLab

Lo scopo essenziale del codice, che si può trovare in appendice, è quello di calcolare la luminosità media di determinate regioni (Region Of Interest) di un frame e determinare se questi eventi erano in contemporanea o meno.

Prima di usare il codice, per l'acquisizione delle webcam abbiamo usato il software "OBS studio" che ci permette di combinare tutti i segnali di più videocamere in un unico frame e quindi registrarlo, oppure creare una webcam virtuale per fare un'analisi in diretta.

Abbiamo scelto di scrivere il codice in MATLAB per la sua intuitività e per la nostra familiarità con questo linguaggio di programmazione.

Innanzitutto abbiamo definito le variabili per indicare cosa e come analizzare:

- NumeroROIs: indica il numero di zone distinte nel frame da analizzare;
- LimiteROI: un intero compreso tra 0 e 255 che indica il limite inferiore della luminosità sopra il quale la ROI viene considerata accesa
- k: se NumeroROIs è maggiore di due si può indicare un intero minore di quest'ultimo che ci permette contare quante volte k ROI erano sopra il limite;

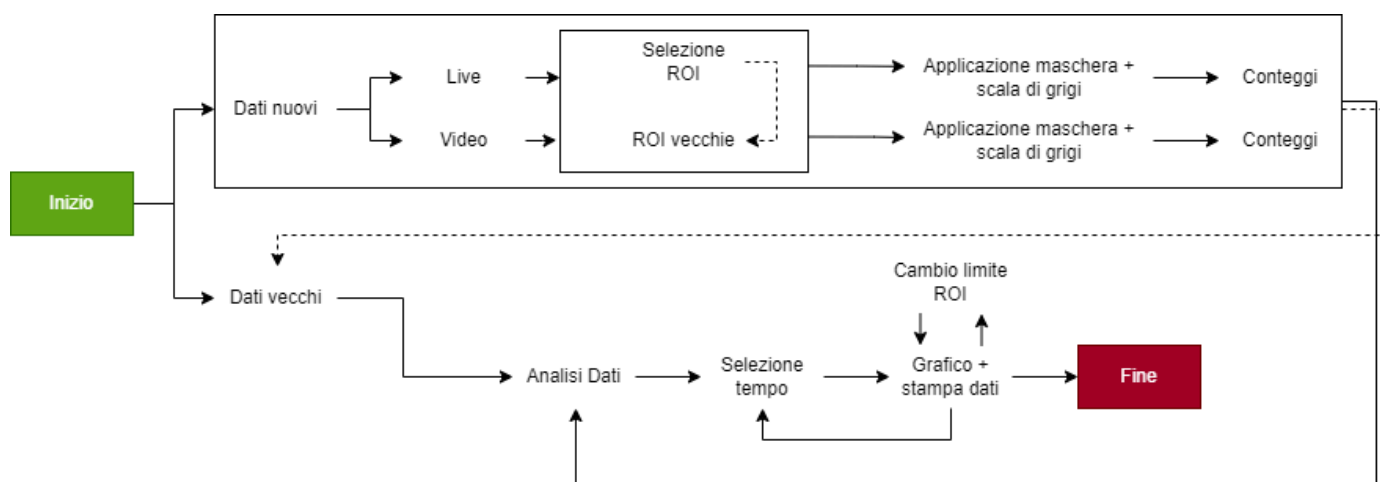
- FileROIs: il percorso del file contenente le coordinate dei vertici delle maschere per indicare le ROI qualora si volessero usare ROI vecchie;
- FileVideo: il percorso del video da analizzare se si usa la modalità video;
- captureDuration: la durata dell'acquisizione (in secondi) se si usa la modalità live;
- DeviceID: il numero identificativo della webcam da usare nella modalità live;
- FrameRate: il numero di frame per secondo che la webcam cattura in modalità live;
- PixelSpenti: un intero che indica il limite inferiore di luminosità sotto il quale un singolo pixel viene considerato spento.

All'avvio del codice viene chiesto se voler usare dati nuovi, quindi analizzare un nuovo video o fare una nuova cattura in diretta, oppure usare dati vecchi, dati precedentemente analizzati dallo stesso codice che vengono salvati su richiesta dell'utente alla fine dei conteggi. Questa seconda opzione è molto utile in quanto potendo analizzare una seconda volta i dati precedentemente esaminati, si risparmia molto tempo per ricompilare il codice.

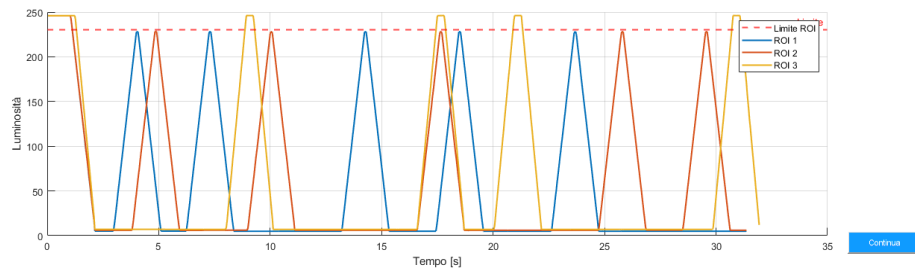
Se si sceglie di creare dati nuovi si dovrà fare anche un'ulteriore scelta, se usare un video oppure se usare un segnale catturato in diretta dalle webcam; così facendo si riesce a risparmiare parecchio tempo soprattutto nelle acquisizioni più lunghe. Successivamente si dovranno individuare la ROI, o manualmente selezionando delle spezzate chiuse sui frame (e in questo caso si potrà anche decidere di salvarle) oppure attraverso un file già creato. A questo punto si procede con l'applicazione delle maschere e con la conversione in scala di grigi (anche se sia in modalità video che in modalità live arriviamo agli stessi dati finali, per come è stato concepito MatLab dobbiamo differenziare il modo in cui il codice esegue questi passaggi nelle due modalità). Per quanto riguarda la conversione in grigi, bisogna analizzare lo spazio di colore usato per la codifica del video o del segnale live, nel nostro caso OBS studio usa la YUV (uno molto diffuso, usato anche dalla televisione satellitare), che è composta da tre componenti con valori da 0 a 255 per ogni pixel: uno per la luminanza (Y) e due per la cromaticità (U e V); per arrivare quindi al nostro risultato ci basta quindi considerare solo la colonna contenente i dati per la luminanza. Per quanto riguarda i conteggi viene fatta la media di tutti i pixel accesi di ogni frame.

Se invece avessimo scelto di usare dati vecchi, avremmo risparmiato tutti questi passaggi e saremmo passati direttamente all'analisi dei dati.

Una volta arrivati a questo punto ci manca solo da individuare l'intervallo di tempo e si genererà il grafico con il tempo sull'ascissa e il limite delle ROI sull'ordinata e nella finestra di comando saranno stampati i dati relativi al numero di volte che tutte le ROI, k ROI e ogni singola ROI hanno superato il limite di luminosità, con la rispettiva frequenza. Schiacciando il tasto continua, inoltre, si può, oltre a terminare il codice, o cambiare di nuovo l'intervallo di tempo oppure cambiare il limite ROI e sarà creato un nuovo grafico e ristampati i dati (queste operazioni sono tutte immediate in quanto si tratta solo di semplici calcoli, al differenza del loop di conteggio che deve analizzare ogni frame).



Di seguito è mostrato un esempio di utilizzo del codice: il grafico e i dati stampati nella finestra di comando. Il grafico è relativo a tutta la durata del video con limite 230 mentre i primi dati sono relativi a tutto il video con limite 200, i secondi tutto il video e limite 230 e i terzi primi 10 secondi e limite 230 (i dati in blu sono quelli relativi al grafico).



```
Analisi iniziata il: 2025-02-25 20:43:14
████████████████████████████████████████████████████████████████████████████████
Durata totale: 31.98 secondi
File video: prova.mp4
Analisi finita il: 2025-02-25 20:44:02
-----
Intervallo di tempo: 0s - 31.976s
tutti ROIs sopra il limite: 37 frames (1.23 seconds)
K ROIs sopra il limite: 47 frames (1.57 seconds)
Numero di volte che tutti i ROIs sono sopra il limite: 1 (Frequenza 0.031273 Hz 1.8764 cpm)
Numero di volte che almeno k ROIs sono sopra il limite: 2 (Frequenza 0.062547 Hz 3.7528 cpm)
Passaggi del limite per singoli ROI:
ROI 1: 88 frames (2.93 seconds, 2.75 Hz, 165.12 cpm)
ROI 1 passaggi Sopra->Sotto: 6 times (0.19 Hz, 11.26 cpm)
ROI 1 passaggi Sotto->Sopra: 5 times (0.16 Hz, 9.38 cpm)
ROI 2: 87 frames (2.90 seconds, 2.72 Hz, 163.25 cpm)
ROI 2 passaggi Sopra->Sotto: 6 times (0.19 Hz, 11.26 cpm)
ROI 2 passaggi Sotto->Sopra: 5 times (0.16 Hz, 9.38 cpm)
ROI 3: 122 frames (4.07 seconds, 3.82 Hz, 228.92 cpm)
ROI 3 passaggi Sopra->Sotto: 5 times (0.16 Hz, 9.38 cpm)
ROI 3 passaggi Sotto->Sopra: 4 times (0.13 Hz, 7.51 cpm)
-----
Nuovo LimiteROI impostato a: 230
Intervallo di tempo: 0s - 31.976s
LimiteROI: 230.00
Tutti ROIs sopra il limite: 33 frames (1.10 seconds)
K ROIs sopra il limite: 33 frames (1.10 seconds)
Numero di volte che tutti i ROIs sono sopra il limite: 1 (Frequenza 0.031273 Hz 1.8764 cpm)
Numero di volte che almeno k ROIs sono sopra il limite: 1 (Frequenza 0.031273 Hz 1.8764 cpm)
Passaggi del limite per singoli ROI:
ROI 1: 33 frames (1.10 seconds, 1.03 Hz, 61.92 cpm)
ROI 1 passaggi Sopra->Sotto: 1 times (0.03 Hz, 1.88 cpm)
ROI 1 passaggi Sotto->Sopra: 0 times (0.00 Hz, 0.00 cpm)
ROI 2: 33 frames (1.10 seconds, 1.03 Hz, 61.92 cpm)
ROI 2 passaggi Sopra->Sotto: 1 times (0.03 Hz, 1.88 cpm)
ROI 2 passaggi Sotto->Sopra: 0 times (0.00 Hz, 0.00 cpm)
ROI 3: 91 frames (3.03 seconds, 2.85 Hz, 170.75 cpm)
ROI 3 passaggi Sopra->Sotto: 5 times (0.16 Hz, 9.38 cpm)
ROI 3 passaggi Sotto->Sopra: 4 times (0.13 Hz, 7.51 cpm)
-----
Intervallo di tempo: 0s - 10s
LimiteROI: 230.00
Tutti ROIs sopra il limite: 33 frames (1.10 seconds)
K ROIs sopra il limite: 33 frames (1.10 seconds)
Numero di volte che tutti i ROIs sono sopra il limite: 1 (Frequenza 0.1 Hz 6 cpm)
Numero di volte che almeno k ROIs sono sopra il limite: 1 (Frequenza 0.1 Hz 6 cpm)
Passaggi del limite per singoli ROI:
ROI 1: 33 frames (1.10 seconds, 3.30 Hz, 198.00 cpm)
ROI 1 passaggi Sopra->Sotto: 1 times (0.10 Hz, 6.00 cpm)
ROI 1 passaggi Sotto->Sopra: 0 times (0.00 Hz, 0.00 cpm)
ROI 2: 33 frames (1.10 seconds, 3.30 Hz, 198.00 cpm)
ROI 2 passaggi Sopra->Sotto: 1 times (0.10 Hz, 6.00 cpm)
ROI 2 passaggi Sotto->Sopra: 0 times (0.00 Hz, 0.00 cpm)
ROI 3: 52 frames (1.73 seconds, 5.20 Hz, 312.00 cpm)
ROI 3 passaggi Sopra->Sotto: 2 times (0.20 Hz, 12.00 cpm)
ROI 3 passaggi Sotto->Sopra: 1 times (0.10 Hz, 6.00 cpm)
-----
```

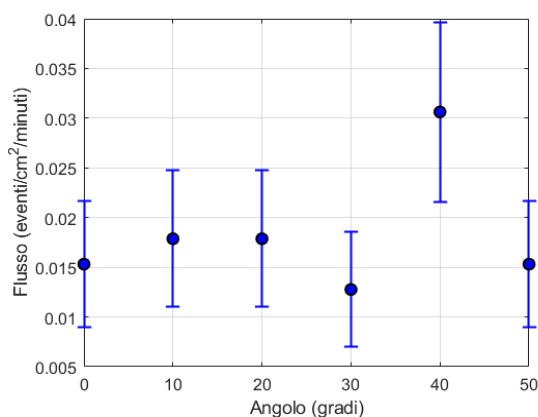
È stato molto utile nella scrittura del codice creare sinteticamente dei video tramite appositi programmi in cui, su uno sfondo nero, comparivano rettangoli bianchi (da uno di essi sono stati presi i dati mostrati sopra) in modo da poter verificare la correttezza dei dati che il codice fornisce. La comodità della modalità “live” è che durante l’acquisizione si riescono ad analizzare i dati in diretta e quindi non si deve aspettare un tempo (pari all’incirca alla stessa durata dell’acquisizione) dopo il termine

dell'esperimento come accade con la modalità video. Quando però abbiamo provato ad usare questa modalità ottenevamo sempre un grafico piatto, quindi con un po' di ricerche abbiamo scoperto che i software di trasmissione video in diretta per questioni di compressione non usano tutto i valori possibili per la luminosità (da 0 a 255) ma limitano il segnale da 16 a 235 tagliando quindi proprio quei dati che eravamo interessati a misurare. Purtroppo non siamo riusciti a trovare soluzioni a questo problema quindi abbiamo sempre usato la modalità video.

Una grande possibilità di questo codice è che è in grado di determinare se due webcam, se messe una sopra l'altra, sono eccitate in contemporanea da una particella e quindi essere sicuri che fosse passata una particella ad alta energia, però non siamo riusciti mai ad ottenere un evento simile. Molto probabilmente possiamo dare la causa di ciò al fatto che, oltre alle piccolissime dimensioni dei sensori (per aumentare l'angolo visivo avremmo potuto avvicinare il più possibile i sensori ma, essendo questi fissati alle rispettive schede, abbiamo stabilito un limite inferiore di 1 cm che ci dà 0.023sr [8]) ad un ispezione più accurata ci siamo accorti che, sebbene le videocamere fossero dello stesso modello, i sensori non erano posizionati nella stessa posizione, ma erano un po' ruotati l'uno rispetto l'altro. Per tutti questi motivi, non siamo riusciti ad usare il codice come avremmo voluto.

Con Matlab siamo riusciti ad ottenere i seguenti dati (al solito l'errore sugli eventi è dato dalla distribuzione di Poisson):

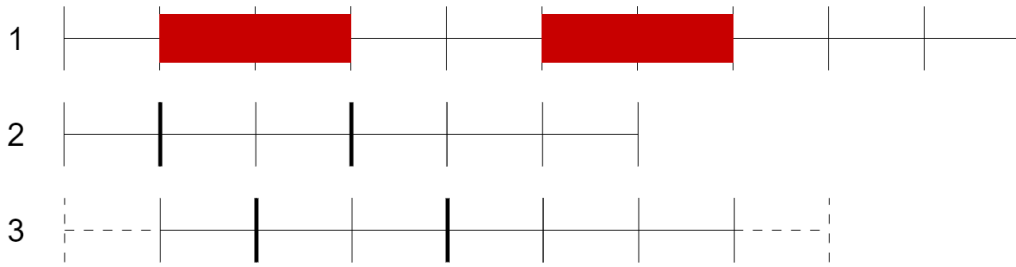
gradi	eventi	errore eventi	flusso (CPM/cm ²)	errore flusso
0	6	2,45	0,92	0,39
10	7	2,65	1,07	0,42
20	7	2,65	1,07	0,42
30	5	2,24	0,77	0,35
40	12	3,46	1,84	0,55
50	6	2,45	0,92	0,39



Problema di combinatoria

Dopo aver analizzato il codice osserviamo come se due eventi sono avvenuti in contemporanea nella stessa webcam, questo non sarà in grado di distinguerli e li conterà come uno singolo. Per risolvere questo problema avremmo dovuto radicalmente cambiare il codice in quanto al posto delle ROI intere, avremmo dovuto trovare nella matrice luminosità i gruppi di caselle con valore superiore al limite e che avessero intorno in tutte le dimensioni caselle sotto il limite e analizzarle. Questo funzione avrebbe di molto complicato la situazione e poiché va ben oltre le nostre capacità tecniche, abbiamo deciso di fare una

scelta consapevole escludendo questi casi. Analizziamo infatti ora la probabilità che non ci siano eventi in contemporanea, supponendo l'emissione dei muoni totalmente casuale facendo riferimento alla figura seguente.



Il nostro problema si riconduce a trovare tutte le possibili combinazioni in cui possiamo sistemare n rettangoli (il numero di eventi) che occupano b spazi adiacenti (i frame che occupa un evento, che assumeremo costanti) in un segmento diviso in a spazi (i frame totali) con l'accortezza di escludere anche il frame precedente e quello successivo ad un evento in quanto sarebbero lo stesso registrati con uno unico (1). Possiamo innanzitutto osservare come non ci interessa tanto la posizione dei singoli rettangoli, che sono tutti della stessa lunghezza, quanto gli intervalli che ci sono tra essi, ci rimangono quindi $a - nb$ spazi (2). Questi intervalli, da come abbiamo detto prima, devono occupare almeno uno spazio ad eccezione del primo e dell'ultimo che possono anche non esserci (il primo evento può cominciare nel primo frame e l'ultimo può finire nell'ultimo frame). Per risolvere il problema possiamo quindi immaginare di aggiungere un frame all'inizio e uno alla fine della registrazione e così la richiesta si trasforma nel trovare in quanti modi possiamo dividere un segmento con $a - nb + 2$ spazi in modo che ogni suddivisione contenga almeno uno spazio (3). Riferendoci ora ai tagli nel segmento (3), che individuano gli intervalli, essi non possono coincidere e dato che possono trovarsi in $(a - nb + 2) - 1 = a - nb - 1$ posizioni e devono essere in numero n , abbiamo:

$$\binom{a - nb + 1}{n}$$

possibili combinazioni. Moltiplichiamo infine questo risultato per tutte le permutazioni possibili degli eventi e abbiamo:

$$\binom{a - nb + 1}{n} n! \quad (a)$$

Per trovare quindi la possibilità che nessun evento avvenga contemporaneamente ad un altro dividiamo (a) per il numero totale di modi in cui possiamo disporre gli eventi, pari a $(a - b + 1)^n$ (basta considerare che ogni evento può cominciare in un qualsiasi frame del video fuorché negli ultimi $b - 1$ in quanto non riuscirebbe a finire in tempo) e si ha:

$$\binom{a - nb + 1}{n} \frac{n!}{(a - b + 1)^n} \quad (b)$$

Poiché nella nostra situazione sarebbe necessario calcolare (b) per un valore molto alto di a ($\sim 10^7$), ciò sarebbe poco praticabile. Possiamo quindi prendere il limite quando a tende ad infinito, tenendo conto che n e b sono diversi ordini di grandezza inferiori rispetto ad a . Si ha:

$$\lim_{a \rightarrow \infty} \binom{a - nb + 1}{n} \frac{n!}{(a - b + 1)^n} = \lim_{a \rightarrow \infty} \binom{a}{n} \frac{n!}{a^n} = \lim_{a \rightarrow \infty} \frac{a(a-1)(a-2) \cdots (a-n) n!}{n! a^n} = \lim_{a \rightarrow \infty} \frac{a^n}{a^n} = 1$$

Possiamo quindi concludere che per video abbastanza lunghi la probabilità di avere eventi in contemporanea tende a zero.

Analisi dei dati

Osservando i dati ricavati dalle varie misurazioni abbiamo cercato di ricavare un modello che ne spiegasse andamento ed eventuali incongruenze.

I nostri ragionamenti sono partiti dall'osservazione: intuitivamente ci aspettiamo che il flusso di raggi cosmici (che provengono per la maggior parte dall'alto) fosse massimo quando il sensore si trovava orizzontale e nullo quando in verticale. Abbiamo quindi assunto che il flusso di raggi cosmici fosse proporzionale al coseno dell'angolo di inclinazione del sensore.

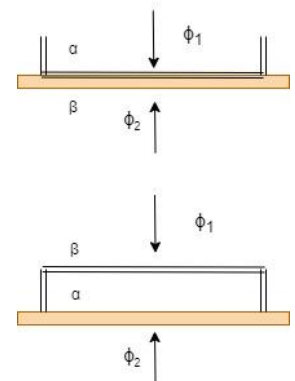
Scrematura dati

In un primo momento, ci siamo concentrati sulla selezione dei dati utili. Abbiamo innanzitutto supposto che i valori a soglia 10 fossero quelli indicativi del flusso di muoni ed escludessero il più possibile l'influenza di altri fattori esterni. Questi valori però, oltre a non essere tutti compatibili gli uni con gli altri, non rispecchiavano il valore che ci aspettavamo, ovvero 1 CPM/cm², come da letteratura [9]. Per questo abbiamo supposto l'esistenza di due coefficienti di efficienza α e β (si veda "modello finale"). Visto inoltre che i valori del flusso diminuivano ad ogni misurazione (si considerino in particolare le misurazioni a soglia 10 eseguite nei giorni 10/02, 13/02, 19/02 con Theremino), ed avendo constatato che l'oscurazione delle webcam si era rovinata, abbiamo deciso di rifare le misurazioni a soglia 10 nell'eventualità che l'oscurazione difettosa le avesse compromesse. Per queste ultime quattro misurazioni abbiamo considerato anche un'inclinazione di 40°.

Modello finale

Il nostro scopo è stato quello di interpretare i dati ottenuti per creare un modello che spiegasse quello che abbiamo misurato. Abbiamo però trovato che il numero dei raggi cosmici dovesse diminuire all'aumentare dell'angolo di inclinazione del sensore seguendo la funzione del $\cos^2(\theta)$ contrariamente a quello che però abbiamo misurato. Quindi abbiamo provato a creare un modello che considerasse separatamente le particelle provenienti dall'alto (i raggi cosmici) e quelle provenienti dalla Terra (la radiazione di fondo).

Siamo partiti dai dati raccolti da Theremino. Abbiamo prima di tutto provato a misurare la radiazione di fondo con un sensore geiger della Vernier (si veda "Apparato sperimentale") con scarsi risultati. Abbiamo quindi introdotto quattro coefficienti di efficienza (numeri compresi tra zero e uno) per indicare quanto il chip influisca sulle misurazioni dei flussi dal basso ϕ_2 e dall'alto ϕ_1 e analizzato i vari dati misurati. Girando la webcam di 180°, abbiamo quindi potuto invertire α e β ma ovviamente non i flussi.



soglia	angolo (θ)	flusso (CPM/cm ²)	coefficienti di efficienza
10	0	A=0.76	α, β
	180	B=0.76	
7	0	C=1.4	γ, δ
	180	D=1.4	

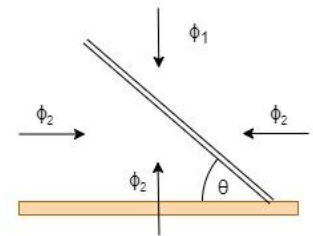
Abbiamo quindi ipotizzato che ogni dato da noi misurato fosse una somma delle combinazioni tra un coefficiente di efficienza e il flusso dall'alto o dal basso e abbiamo impostato il seguente sistema

$$\begin{cases} \alpha\phi_1 + \beta\phi_2 = A \\ \alpha\phi_2 + \beta\phi_1 = B \\ \gamma\phi_1 + \delta\phi_2 = C \\ \gamma\phi_2 + \delta\phi_1 = D \end{cases} \quad (a)$$

Come mostrato dalla fonte [10] esiste una soglia di luminosità sotto la quale il sensore non cattura più particelle ma conta eventi anomali che non possono essere considerati attendibili, quando siamo su questa soglia, quindi possiamo assumere di star contando tutte le particelle di fondo. Nel nostro caso abbiamo trovato che questa soglia è 7 e quindi possiamo assumere delta uguale a 1. Inoltre assumiamo ϕ_1 uguale a 1 CPM/cm² come da letteratura [9].

Abbiamo prima risolto il sistema formato dalle ultime due equazioni in quanto, per quanto detto sopra, le uniche incognite erano ϕ_2 e γ e ottenuto le coppie di risultati (0.4, 1) e (1, 0.4). Provando però a reinserire i valori della seconda coppia nelle prime due equazioni abbiamo ottenuto un sistema indeterminato ($\alpha + \beta = 0.76$ e $\alpha + \beta = 0.76$), possiamo quindi concludere che $\phi_2 = 0.4$ CPM/cm² e $\gamma = 1$. Quindi inserendo questi dati in (a) troviamo, infine, $\alpha = \beta = 0.54$.

Dati i risultati ottenuti proviamo quindi ad analizzare anche i dati con la webcam inclinata di 40°. Se supponiamo che il flusso di particelle cosmiche segua la legge del coseno al quadrato, mentre che il flusso di fondo non lo segua ma interagisca con la proiezione dell'area del sensore sul piano orizzontale e due volte con quello verticale (il fondo colpisce il sensore da destra, da sinistra e dal basso) si ha la seguente legge, indicando con E il flusso ideale:



$$E = \alpha\phi_1\cos^2\theta + \beta\phi_2(\cos\theta + 2\sin\theta) = 0.76\text{cpm}/\text{cm}^2 \quad (b)$$

che rientra nell'intervallo di incertezza della media dei due valori misurati:

$$\frac{0.82 + 0.71}{2} = (0.77 \pm 0.07)\text{cpm}/\text{cm}^2$$

Incoraggiati dai valori ottenuti abbiamo deciso di eseguire un'ulteriore prova, inclinando la webcam di un angolo $\theta=90^\circ$. Secondo il nostro modello il flusso ottenuto dovrebbe corrispondere al doppio del flusso del fondo, in quanto, secondo la legge del $\cos^2(\theta)$, a 90° non si dovrebbero rilevare i muoni, ed effettivamente i dati sperimentali sono compatibili con il nostro modello. Le particelle rilevate con $\theta=90^\circ$ sono 26 per un periodo di tempo di 18269 s, il flusso di particelle ha perciò un valore di $(0,8\pm0,1)$ CPM/cm², ovvero il valore che ci aspettavamo.

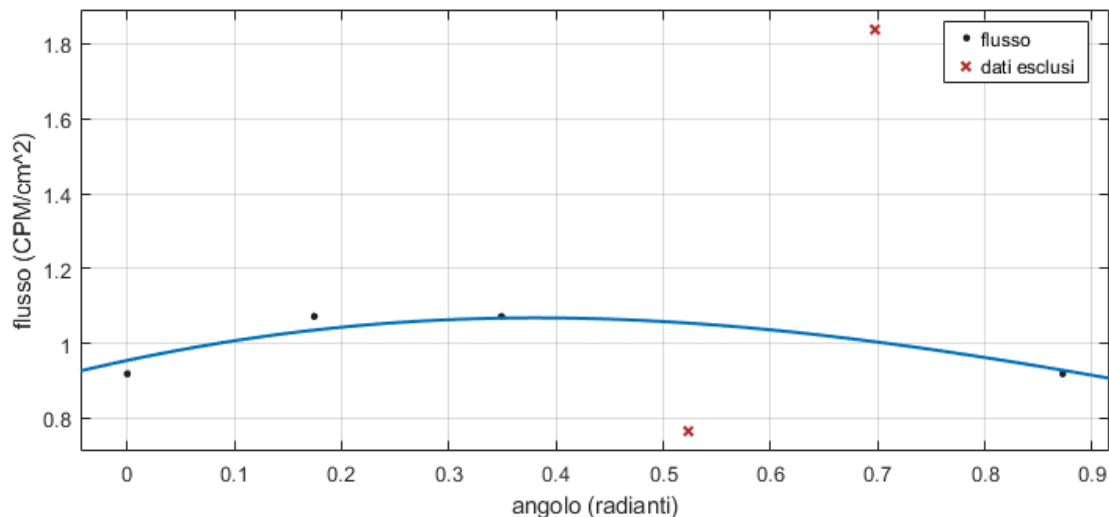
Per quanto riguarda i dati ottenuti con matlab, purtroppo per questioni di tempo non siamo riusciti ad eseguire un'acquisizione con la webcam capovolta, quindi non abbiamo tutti i dati per risolvere il sistema (a). Nonostante ciò abbiamo provato a fare un curve fitting del flusso in funzione dell'angolo, tenendo come parametri liberi i coefficienti $a = \alpha\phi_1$ e A. La funzione per il fit è quella che si ottiene combinando la prima equazione di (a) con (b), ovvero $f(a, A, \theta) = a(\cos^2\theta - \cos\theta - 2\sin\theta) + A(\cos\theta + 2\sin\theta)$

Dal fit otteniamo: $a = 0.6514$ CPM/cm², $A=0.9545$ CPM/cm² con $r^2=0.88$ (abbiamo deciso di escludere due dati perché potrebbe essersi verificato un problema con il LED della webcam). Per quanto riguarda il valore

di A, esso è compatibile entro la barra d'errore del dato sperimentale misurato da noi e che risulta (0.92 ± 0.39) CPM/cm² nella tabella presentata prima del problema di combinatoria. Per quanto riguarda il parametro a , notiamo innanzitutto che l'errore sui conteggi della tabella è alto, in percentuale, perché non avevamo ancora chiaro che fosse necessario fare acquisizioni di ore, come fatto poi con Theremino. Se teniamo conto della media degli errori in percentuale che ricaviamo dalla tabella presentata prima del problema di combinatoria, otteniamo un errore dei conteggi che varia da un 30% ad un 45%. Volendo essere pessimisti, assegnamo l'errore minimo al coefficiente del fit ed $a = 0.6514$ diventa:

$$a = (0.7 \pm 0.2) \text{ CPM/cm}^2$$

Il risultato del fit risulta quindi compatibile con il valore ottenuto usando i dati ricavati dal sistema di Theremino, ovvero 0.54. Viceversa, possiamo dire che noto il coefficiente del fit, il valore del flusso della radiazione di fondo e del flusso di muoni sono noti al più con una precisione del 30%.



Conclusioni

Come mostrato, il nostro esperimento alla fine non si è focalizzato sull'analisi della velocità dei muoni. Le principali motivazioni sono state le incongruenze rilevate tra i vari dati e con i valori della letteratura. Ci siamo infatti concentrati sulla ricerca di un modello che spiegasse i nostri dati, servendoci di ulteriori misurazioni e supposizioni (ad esempio la presenza di coefficienti di efficienza e l'influenza dell'effetto dell'angolo di inclinazione del sensore). Siamo arrivati alla conclusione che il nostro chip ha due coefficienti di efficienza che dipendono dalla soglia impostata (a soglia 10, $\alpha = \beta = 0.54$). Questi coefficienti spiegano il motivo per cui il flusso di muoni rilevato dalle misurazioni è diverso da quello che ci si aspetta. Per quanto riguarda l'effetto dell'angolo, abbiamo verificato come il sensore rilevi il massimo flusso di muoni quando posizionato orizzontalmente e come la rilevazione del flusso della radiazione di fondo dipenda dall'angolo di inclinazione del sensore. Quando il sensore è posto perpendicolare all'orizzontale, infine, non rileva muoni, andando a confermare sperimentalmente la legge del $\cos^2(\theta)$. Confrontando infine i dati di due programmi differenti, possiamo dire che abbiamo misurato un flusso di muoni che è pari a quello presente in letteratura a meno di un errore relativo del 30%, ovvero è pari a quello con cui l'ultimo valore del fit dato da Matlab è compatibile con i dati ricavati con Theremino.

Bibliografia e sitografia

1. Darja Nonats, Dall'universo alla terra, Meridiana 224 05-06 2013;

2. <https://www.astroparticelle.it/muon-detector.asp> ;
3. <https://physicsopenlab.org/2016/05/18/rivelatore-di-particelle-diy-basato-su-webcam/> ;
4. <https://it.mathworks.com/help/images/measure-distances-in-images.html> ;
5. <https://www.baumer.com/> ;
6. https://it.wikipedia.org/wiki/Distribuzione_di_Poisson;
7. https://physicsopenlab.org/wp-content/uploads/2021/07/Theremino_ParticleDetector_1.1.zip ;
8. https://en.wikipedia.org/wiki/Solid_angle#Pyramid ;
9. Autran, J.L. et al. (2018). Characterization of atmospheric muons at sea level using a cosmic ray telescope. *Nuclear Instruments. and Methods in Physics Research, A*, **903**, pp. 77-84 ;
10. <https://indico.cern.ch/event/844366/contributions/3545400/attachments/1900583/3137140/ESR7-FinalConference.pdf> .

Appendice codice

```
%Variabili -----
NumeroROIs = 2;
LimiteROI = 1.9; %0<=LimiteROI<=255
k = 2;
FileROIs = 'zROIsSalvate.mat';
FileVideo = 'file.mp4';
captureDuration = 1; % (secondi)
DeviceID = 1;
FrameRate = 30;
PixelSpenti = 0;
%-----
fprintf('Analisi iniziata il: %s\n', string(datetime('now', 'Format', 'yyyy-MM-dd HH:mm:ss')));
CambioTempo = true;
Grafico = [];
Scelta2 = questdlg('Usare dati...', '', 'Vecchi', 'Nuovi', 'Vecchi');
Datinuovi = false;
switch Scelta2
    case 'Nuovi'
        Datinuovi = true;
    case 'Vecchi'
        Datinuovi = false;
end
if Datinuovi
    Scelta = questdlg('Modalita da usare', '', 'Video', 'Live', 'Video');
    SalvaVariabili = questdlg('Salvare le matrici e variabili', '', 'Si', 'No', 'No');
    ModalitaVideo = false;
    switch Scelta
        case 'Video'
            ModalitaVideo = true;
        case 'Live'
            ModalitaVideo = false;
    end
end
if ModalitaVideo
    Video = VideoReader(FileVideo);
    FrameRate = Video.FrameRate;
    NumeroFrames = Video.NumFrames;
    TempoFrame = (1:NumeroFrames) / Video.FrameRate;
    Frame = read(Video, 1);
    Durata = Video.Duration;
    Tempo = (1:NumeroFrames) / Video.FrameRate;
else
    imgreset;
    VideoDevice = imaqhwinfo;
    AdaptorName = 'winvideo';
    Risoluzione = imaqhwinfo(AdaptorName).DeviceInfo(DeviceID).SupportedFormats{1};
    Video = videoinput(AdaptorName, DeviceID, Risoluzione);
    triggerconfig(Video, 'manual');
    set(Video, 'FramesPerTrigger', 1);
    set(Video, 'TriggerRepeat', Inf);
    start(Video);
    NumeroFrames = captureDuration * FrameRate;
    TempoFrame = (1:NumeroFrames) / FrameRate;
    Frame = getsnapshot(Video);
    Durata = captureDuration;
end

MatriceROI = cell(NumeroROIs, 1);
RisultatiLuminosita = zeros(NumeroFrames, NumeroROIs);
ConteggioLimite = zeros(1, NumeroROIs);
PassaggiSopraSotto = zeros(NumeroROIs, 2);
Luminosita = zeros(NumeroFrames, NumeroROIs);
PixelAccesiROI = zeros(NumeroFrames, NumeroROIs);

UsaFileROIs = false;
```



```

function AnalisiROIs(NumeroROIs, Luminosita, LimiteROI, k, Durata, FrameRate)
    PassaggiSopraSotto = zeros(NumeroROIs, 2);
    for i = 1:NumeroROIs
        SopraLimite = Luminosita(:, i) > LimiteROI;
        Passaggi = diff(SopraLimite);
        PassaggiSopraSotto(i, 1) = sum(Passaggi == 1);
        PassaggiSopraSotto(i, 2) = sum(Passaggi == -1);
    end
    PassaggiTotali = sum(all(Luminosita > LimiteROI, 2));
    PassaggiParziali = sum(sum(Luminosita > LimiteROI, 2) >= k);
    TuttiSopraLimite = all(Luminosita > LimiteROI, 2);
    AlmenoKROIsSopraLimite = sum(Luminosita > LimiteROI, 2) >= k;
    NumTuttiSopra = sum(diff([0; TuttiSopraLimite]) == 1);
    NumerokSopra = sum(diff([0; AlmenoKROIsSopraLimite]) == 1);
    FreqTuttiSopra = NumTuttiSopra / Durata;
    FreqkSopra = NumerokSopra / Durata;
    fprintf('LimiteROI: %.2f\n', LimiteROI);
    fprintf('Tutti ROIs sopra il limite: %d frames (%.2f seconds)\n', PassaggiTotali, sum(TuttiSopraLimite) / FrameRate);
    fprintf('K ROIs sopra il limite: %d frames (%.2f seconds)\n', PassaggiParziali, sum(AlmenoKROIsSopraLimite) / FrameRate);
    disp(['Numero di volte che tutti i ROIs sono sopra il limite: ', num2str(NumTuttiSopra), ...
        ' (Frequenza ', num2str(FreqTuttiSopra), ' Hz ', num2str(FreqTuttiSopra * 60), ' cpm)']);
    disp(['Numero di volte che almeno K ROIs sono sopra il limite: ', num2str(NumerokSopra), ...
        ' (Frequenza ', num2str(FreqkSopra), ' Hz ', num2str(FreqkSopra * 60), ' cpm)']);
    disp('Passaggi del limite per singoli ROI:');
    for i = 1:NumeroROIs
        fprintf('ROI %d: %d frames (%.2f seconds, %.2f Hz, %.2f cpm)\n', i, ...
            sum(Luminosita(:, i) > LimiteROI), sum(Luminosita(:, i) > LimiteROI) / FrameRate, ...
            sum(Luminosita(:, i) > LimiteROI) / Durata, (sum(Luminosita(:, i) > LimiteROI) / Durata) * 60);
        fprintf('ROI %d passaggi Sopra->Sotto: %d times (%.2f Hz, %.2f cpm)\n', i, ...
            PassaggiSopraSotto(i, 2), PassaggiSopraSotto(i, 2) / Durata, (PassaggiSopraSotto(i, 2) / Durata) * 60);
        fprintf('ROI %d passaggi Sotto->Sopra: %d times (%.2f Hz, %.2f cpm)\n', i, ...
            PassaggiSopraSotto(i, 1), PassaggiSopraSotto(i, 1) / Durata, (PassaggiSopraSotto(i, 1) / Durata) * 60);
    end
end
while true
    if ~isempty(Grafico) && isvalid(Grafico)
        close(Grafico);
    end
    if CambioTempo
        Domanda = {'Inserisci il tempo di inizio (secondi):', 'Inserisci il tempo di fine (secondi):'};
        RispostaTempo = inputdlg(Domanda, '', [1 50], {'0', num2str(Durata)});
    else
        end
        InizioGrafico = max(0, str2double(RispostaTempo{1}));
        FineGrafico = min(Durata, str2double(RispostaTempo{2}));
        InizioFrameGrafico = max(1, floor(InizioGrafico * FrameRate) + 1);
        FineFrameGrafico = min(NumeroFrames, ceil(FineGrafico * FrameRate));
        TempoGrafico = Tempo(InizioFrameGrafico:FineFrameGrafico);
        LuminositaGrafico = Luminosita(InizioFrameGrafico:FineFrameGrafico, :);
        LuminositaIntervallo = Luminosita(InizioFrameGrafico:FineFrameGrafico, :);
        if CambioTempo
            disp(['Intervallo di tempo: ', num2str(InizioGrafico), 's - ', num2str(FineGrafico), 's']);
            AnalisiROIs(NumeroROIs, LuminositaIntervallo, LimiteROI, k, (FineGrafico - InizioGrafico), FrameRate);
            disp('-----');
        else
            end
            Grafico = figure;
            set(Grafico, 'Color', 'White', 'Units', 'Normalized', 'OuterPosition', [0, 0.1, 1, 0.6]);
            hold on;
            legend('show');
            grid on;
            xlabel('Tempo [s]');
            ylabel('Luminosit ');
            colors = lines(NumeroROIs);
            yline(LimiteROI, '--r', 'Limite', 'LineWidth', 1.5, 'DisplayName', 'Limite ROI');
            for i = 1:NumeroROIs
                plot(TempoGrafico, LuminositaGrafico(:, i), 'Color', colors(i, :), 'LineWidth', 1.5, 'DisplayName', ['ROI ' num2str(i)]);
            end
            hold off;
            uicontrol('Style', 'pushbutton', 'String', 'Continua', 'Position', [1250, 20, 100, 30], 'BackgroundColor', [0.07, 0.62, 1.00], ...
                'ForegroundColor', [1, 1, 1], 'Callback', 'uiresume(gcf)');
            uiwait(Grafico);
            if ~isvalid(Grafico)
                break;
            end
            Risposta = questdlg('Vuoi cambiare LimiteROI o l intervallo di tempo?', '', 'Cambia LimiteROI', 'Cambia Intervallo', 'Fine', 'Cambia Intervallo');
            switch Risposta
                case 'Cambia LimiteROI'
                    newLimiteROI = inputdlg('Nuovo LimiteROI', '', [1 50], {num2str(LimiteROI)});
                    if ~isempty(newLimiteROI)
                        LimiteROI = str2double(newLimiteROI{1});
                        disp(['Nuovo LimiteROI impostato a: ', num2str(LimiteROI)]);
                        LuminositaIntervallo = Luminosita(InizioFrameGrafico:FineFrameGrafico, :);
                        disp(['Intervallo di tempo: ', num2str(InizioGrafico), 's - ', num2str(FineGrafico), 's']);
                        AnalisiROIs(NumeroROIs, LuminositaIntervallo, LimiteROI, k, (FineGrafico - InizioGrafico), FrameRate);
                        disp('-----');
                    end
                    CambioTempo = false;
                case 'Cambia Intervallo'
                    CambioTempo = true;
                    continue;
                case 'Fine'
                    break;
            end
        end
    end
end

```

MESSAGGERI DEL COSMO

RELAZIONE DEL DOCENTE

Quest'anno ho deciso di non imporre al gruppo di ragazzi l'argomento da affrontare, ma di sceglierlo confrontandomi con loro. Purtroppo questa scelta ha generato piccoli attriti nel gruppo che hanno rallentato la scelta della strada da intraprendere. **Con molta fatica abbiamo deciso di dedicarci alla rilevazione dei muoni, con l'ambizione di poter misurare magari la differenza di flusso al variare dell'altitudine.** Questo non è stato possibile, ma il percorso ci ha portato a ragionare su aspetti strutturali del metodo scientifico.

Una volta scelto l'argomento, mi sono relazionato con i ragazzi come farebbe un Principal Investigator (PI) di un gruppo di ricerca. Ho quindi guidato le discussioni, assegnato compiti, seguito in parte le idee scaturite dal gruppo, ed in parte le ho guidate. Ho dato vari input sulla ricerca bibliografica iniziale. La prima idea era quella di costruire un rivelatore che sfruttasse il metodo della coincidenza attraverso l'accensione di tubi fluorescenti (neon), la seconda quella di sfruttare le webcam che avevamo in eredità dal periodo del Covid19. La necessità di usare l'alta tensione per il primo caso ha rallentato enormemente la costruzione del primo apparato ed ho quindi spinto perché il gruppo si focalizzasse sull'uso delle webcam. Come accaduto in passato, i siti consultati suggerivano si trattasse di una strada non troppo irta di ostacoli. Ovviamente così non è stato.

Dal punto di vista didattico, due cose sono emerse molto chiaramente. La prima è stata l'importanza di studiare lo strumento di misura, conoscendone il principio di funzionamento e i suoi limiti. Questo è emerso per tentativi ed errori, imbastendo l'esperimento e modificando di volta in volta l'approccio man mano che emergevano nuove informazioni sullo strumento stesso (le webcam e il CMOS). L'esplorazione di come funzionano le webcam ci ha portato a misurare l'area del CMOS e per constatare che le misure trovate sui siti non erano standard, che le webcam sembrano uguali ma sono in realtà molto differenti. Infine, mentre i tempi di acquisizione iniziali variavano tra i 5 minuti e l'ora, le acquisizioni finali sono tutte tra le 8 e le 24 ore!

L'altro punto importante è stato capire come interpretare i dati sperimentali che avevamo, chiedendoci cosa avessimo misurato e come sfruttare i primi dati per riprogrammare l'esperimento. In questo frangente ho discusso molto con i ragazzi, proponendo un modello interpretativo che ho però sottoposto alle loro critiche. La discussione è stata proficua, perché ci ha permesso di riprogrammare le acquisizioni con la webcam ed ottenere anche delle conferme del modello interpretativo dei dati.

L'eterogeneità ha spinto uno dei componenti a tentare di scrivere un codice per interfacciarsi con il CMOS, mentre altri hanno proposto l'utilizzo di un software libero (Theremino). Il confronto di questi due metodi ha prodotto un confronto all'interno del gruppo e gestito dai ragazzi stessi.

Lo scopo iniziale di misurare il flusso di muoni è stato raggiunto anche se per una via più tortuosa ed indiretta. Questo percorso ha prodotto uno studio a mio avviso dell'impiego delle webcam che non è presente in letteratura con questo dettaglio. Infine, la sfida di costruire anche un rivelatore con i neon è ancora sulla nostra to-do list e il lavoro fatto ha stimolato il gruppo a continuare l'indagine anche dopo la consegna delle tesine.