

Università degli Studi dell'Insubria
Dipartimento di Scienze e Alta Tecnologia
Corso di Laurea Triennale in Fisica

Imaging di una sorgente con rivelatore MIMOTERA

Erica Brisigotti & Ylenia Mascolo



Esame di Laboratorio di Fisica I
Docente: Prof. Dott. Massimo Luigi Maria Caccia
Assistante di Laboratorio: Dott.ssa Samuela Lomazzi
Anno Accademico 2018-2019

Indice

1	Introduzione	3
1.1	L'Americio e il decadimento α	3
1.2	Il rivelatore MIMOTERA	4
1.3	I dati	6
2	Analisi dati	7
2.1	Visualizzazione preliminare	7
2.2	Equalizzazione	8
2.2.1	Sottrazione di piedistallo per quadrante	9
2.2.2	Sottrazione di piedistallo per pixel	10
2.3	<i>Quantum imaging</i> e individuazione dei <i>cluster</i>	13
3	Conclusioni	16
4	Appendice	17
4.1	Media e relativa incertezza	17
4.2	Codice per individuazione dei <i>cluster</i>	17
4.3	Analisi supplementare del <i>set</i> di dati Stuzzicadenti	20
5	Fonti	23

1 Introduzione

L'obiettivo dell'esperimento consiste nella ricostruzione del profilo spaziale della radioattività α di una sorgente di americio-241 posizionata in prossimità di un rivelatore MIMOTERA. Tale profilo spaziale non è del tutto caratteristico della sorgente α ma risulta influenzato da forma e dimensioni del collimatore interposto tra il rivelatore e la sorgente: di conseguenza, quando si menzionerà di seguito la forma della sorgente si intenderà quella del collimatore.

In particolare sono state analizzate due matrici di dati (A e B) corrispondenti alla lettura alternata del segnale da parte dei circuiti A e B. Entrambe verranno di seguito sottoposte alle medesime analisi: le due strategie implementate consistono nell'equalizzazione e nell'individuazione di *cluster* (gruppi di pixel adiacenti). In entrambi i casi, per indagare quantitativamente la forma della sorgente è stato necessario discriminare il profilo tramite l'analisi delle proiezioni.

1.1 L'Americio e il decadimento α

L'americio (^{95}Am) è un elemento metallico radioattivo sintetico della famiglia degli attinidi, che si ottiene bombardando il plutonio (^{94}Pu) con neutroni. È stato il quarto elemento transuranico ad essere scoperto e prende il nome dall'America (in analogia con l'europio): fu sintetizzato per la prima volta da Glenn Seaborg, Leon O. Morgan, Ralph A. James e Albert Ghiorso nel 1944 nel *Metallurgical Laboratory* dell'Università di Chicago.

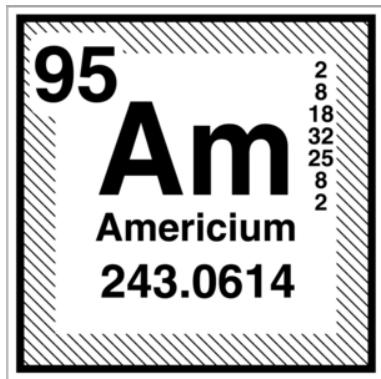


Figura 1: Proprietà chimiche dell'americio.

Sono noti 18 diversi isotopi radioattivi dell'Americio, di cui i più stabili sono ^{243}Am (con un'emivita di 7370 anni) e ^{241}Am (432,2 anni). Quest'ultimo è l'isotopo più comune in natura e quello più diffuso nelle scorie nucleari.

Inoltre, l'americio-241 rappresenta anche il primo isotopo radioattivo di americio sintetizzato: il gruppo del *Metallurgical Laboratory* preparò ^{241}Am sottponendo ^{239}Pu a successive reazioni di cattura neutronica in un reattore nucleare. Questo produsse prima ^{240}Pu e poi ^{241}Pu che a sua volta si convertì in ^{241}Am tramite un decadimento β . Tale isotopo non viene quindi sintetizzato direttamente dall'uranio (materiale più comune in un reattore) siccome è un suo isotopo padre.

L'americio-241 decade principalmente α , emettendo anche raggi gamma:



Le energie di decadimento α sono 5.486 MeV per l'85% del tempo, 5.443 MeV per il 13% del tempo e 5.388 MeV per il restante 2%. Invece, l'energia dei raggi γ è per la maggior parte 59.5409 keV.

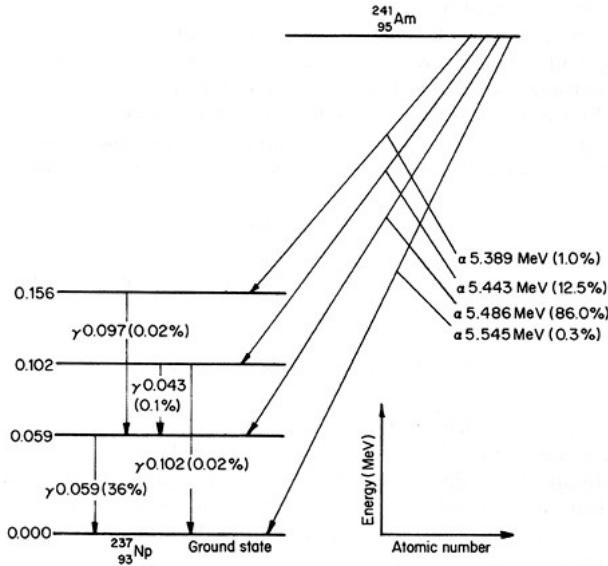


Figura 2: Schema di decadimento di americio-241.

L'americio-241 è l'unico isotopo sintetico utilizzato comunemente nelle abitazioni, nei rilevatori di fumo (a ionizzazione) che impiegano $^{241}\text{AmO}_2$ (biossalido di americio-241) come fonte di radiazioni ionizzanti; inoltre è un potenziale carburante per generatori termoelettrici a radioisotopi a lunga durata.

Tale isotopo risulta però estremamente tossico e radioattivo: sebbene le particelle α possano essere bloccate da un foglio di carta, l'ingerimento di emettitori α costituisce un serio rischio per la salute umana. L'americio e i suoi isotopi sono anche molto tossici, dal punto di vista chimico, sotto forma di tossicità da metalli pesanti per i quali è stato costituito un limite massimo consentito pari a $0.03 \mu\text{Ci}$ (1110 Bq). Inoltre, anche pochi grammi di americio-241 emettono radiazione γ talmente intensa da richiedere l'adozione di misure radio-protettive da parte degli individui che maneggiano la sorgente.

1.2 Il rivelatore MIMOTERA

Il rivelatore MIMOTERA nasce nell'ambito della collaborazione SUCIMA¹ (*Silicon Ultra fast Cameras for electron and gamma sources In Medical Application*), un progetto supportato dalla Commissione Europea mirato all'esecuzione di studi dosimetrici che, nel caso particolare del MIMOTERA, sono stati condotti con rivelatori a substrato in silicio.

Il MIMOTERA (*Minimum Ionizing MONolithic active pixel sensor for TERA foundation*) è infatti un sensore monolitico a pixel attivi ad alta frequenza di fotogrammi, originariamente ideato per la profilometria di fasci di particelle e per la misura della relativa intensità.

¹Successivamente al progetto SUCIMA tale rivelatore è utilizzato anche in molti altri progetti, tra cui l'esperimento AEgis, condotto al CERN e volto a verificare il Principio dell'Equivalenza Debole per l'antimateria e diversi lavori nel campo dell'adroterapia.

Le caratteristiche tecniche principali del MIMOTERA sono le seguenti:

- ha una struttura monolitica, retro-illuminata grazie alla presenza di una sottile finestra di ingresso che garantisce sensibilità anche a particelle ionizzanti di bassa energia (fino a circa 20 keV);
- la capacità complessiva di deposito energetico può raggiungere fino a 30 MeV/pixel;
- è presente un'area sensibile pari a $17 \times 17 \text{ mm}^2$ composta da una matrice 112×112 di pixel quadrati (rispettivamente di passo $153 \mu\text{m}$ ciascuno), organizzati in quattro sotto-matrici da 28×112 pixel, rappresentate di seguito:

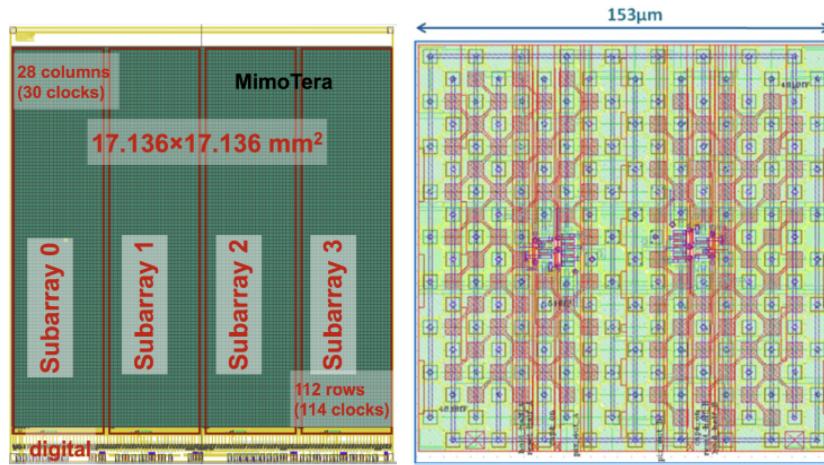


Figura 3: Schematizzazione del rivelatore e layout di un singolo pixel.

- ciascun pixel ha un'architettura basata sulla presenza di due circuiti in grado di garantire uno stato simultaneo di lettura e scrittura permanente (*dead-timeless functionality*), legati a celle di memoria gemelle denominate A e B: mentre il circuito A integra la carica depositata, il circuito B si occupa della lettura. Al termine dell'acquisizione dei dati derivanti da un *frame*, il ruolo di A e B si inverte, permettendo al sensore uno stato di lettura/scrittura permanente senza tempo morto. Nella modalità utilizzata per la presa dati, un *frame* corrisponde ad un intervallo di tempo di circa 10^{-4} s.

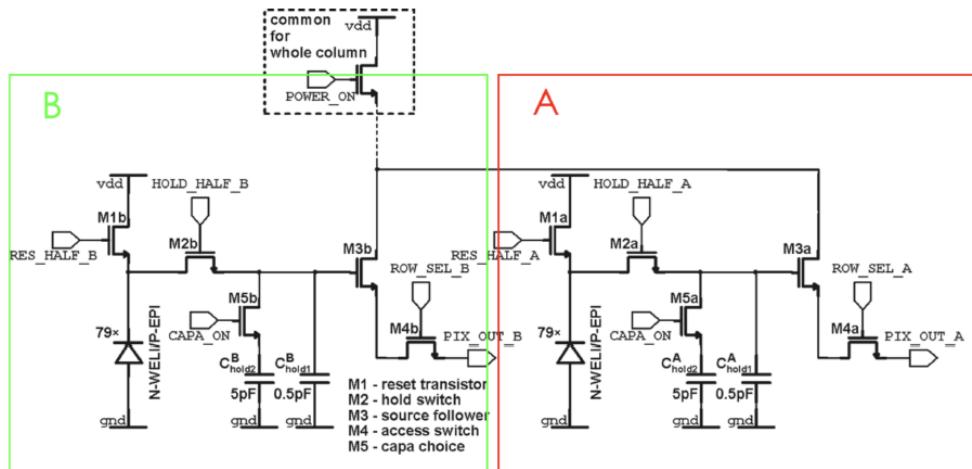


Figura 4: Struttura interna di ciascun pixel, con evidenziati i due circuiti gemelli A e B.

1.3 I dati

Il rivelatore MIMOTERA è stato posizionato a ridotta distanza dalla sorgente circolare di Americio-241 la cui emissione è stata concentrata su una porzione del rivelatore stesso tramite l'utilizzo di un collimatore (di centro e dimensioni incognite).

I dati analizzati sono stati forniti sotto forma di due matrici tridimensionali (corrispondenti alla lettura alternata dei segnali da parte dei circuiti A e B) costituite da 4000 *frame* (fotogrammi) assimilabili a matrici 112×112 .

La seconda metà di questi fotogrammi non contiene segnale (si tratta di matrici nulle) e sono stati pertanto esclusi dall'analisi riducendo i dati esaminati a matrici tridimensionali $112 \times 112 \times 2000$.

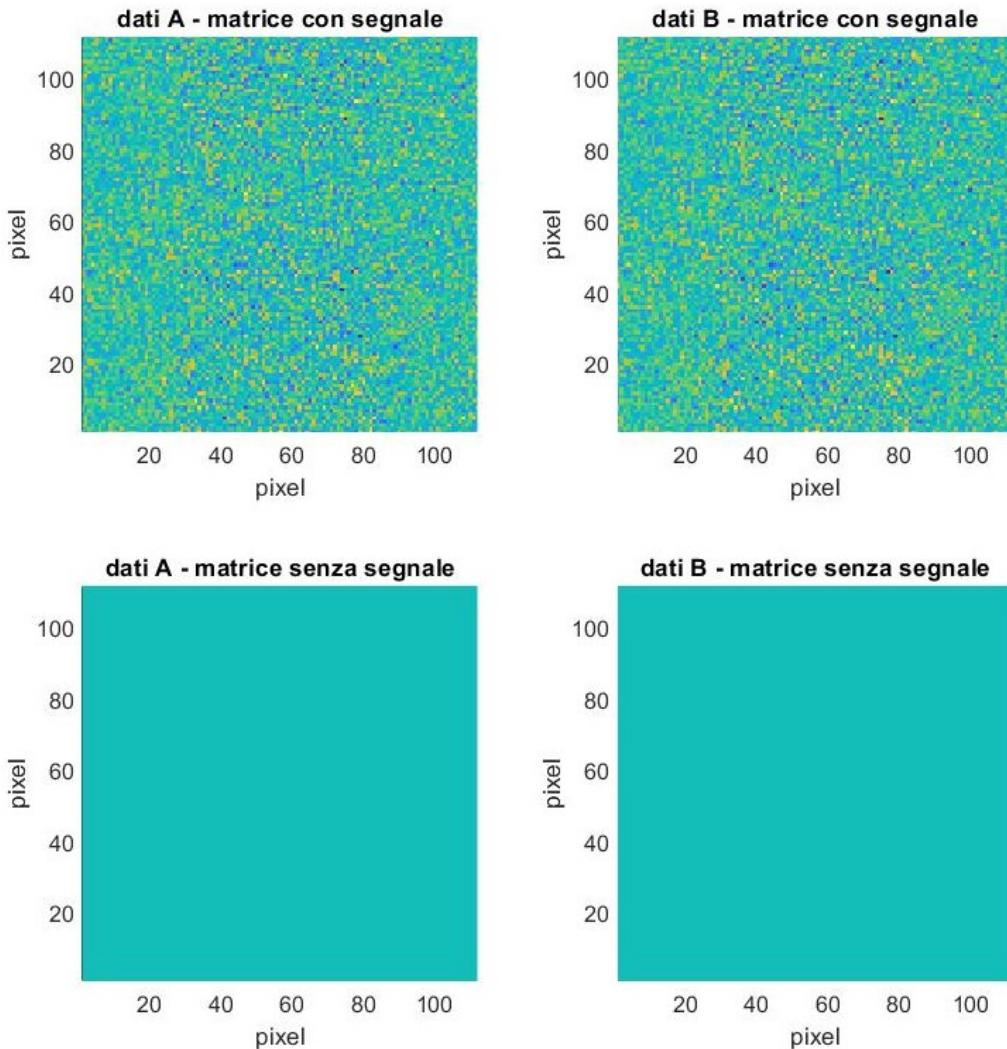


Figura 5: Esempi di *frame* con segnale (corrispondenti a fotogrammi della prima metà dei dati) e senza segnale (appartenenti alla seconda metà), per le matrici di dati A e B.

Inoltre, nell'analisi a seguire sarà sempre impiegato il modulo dei dati registrati nelle matrici $112 \times 112 \times 2000$ in modo tale che, combinando i vari *frame*, i relativi dati possano elidersi vicendevolmente: tali dati quantificano infatti l'energia depositata dalla particella che può essere registrata, in modo equiprobabile, come positiva o negativa.

2 Analisi dati

Date due matrici (A e B) di lettura alternata del segnale, per identificare la posizione della sorgente e le sue dimensioni, sono state impiegate diverse strategie di analisi: prima di tutto è stata trattata la questione del rumore di fondo (che si presenta sotto forma di piedistallo) in modo tale da distinguere correttamente le intensità di energia relative alla sorgente.

Tale piedistallo è stato individuato, nella fase di equalizzazione, prima per quadrante e successivamente per singolo pixel.

La strategia di equalizzazione non ha però restituito un'immagine sufficientemente nitida (pulita dal rumore) della sorgente: si è quindi deciso di mettere in pratica un metodo più raffinato che sfrutta le soglie di rumore trovate nella fase di equalizzazione di singolo pixel per trasformare la matrice tridimensionale originaria in una matrice tridimensionale di booleani.

Questa ha permesso di identificare i *cluster*, ovvero gruppi di pixel che hanno registrato il medesimo impatto di una particella α , e i relativi baricentri da cui si identificheranno le dimensioni e la posizione della sorgente rispetto al rivelatore.

2.1 Visualizzazione preliminare

Come è possibile osservare nella figura (5) i singoli fotogrammi non forniscono indicazioni significative per l'individuazione della sorgente, pertanto si è deciso di sommare il segnale presente nei vari *frame* in modo da ottenere informazioni sul fenomeno complessivo.

I risultati ottenuti sono riportati a seguire:

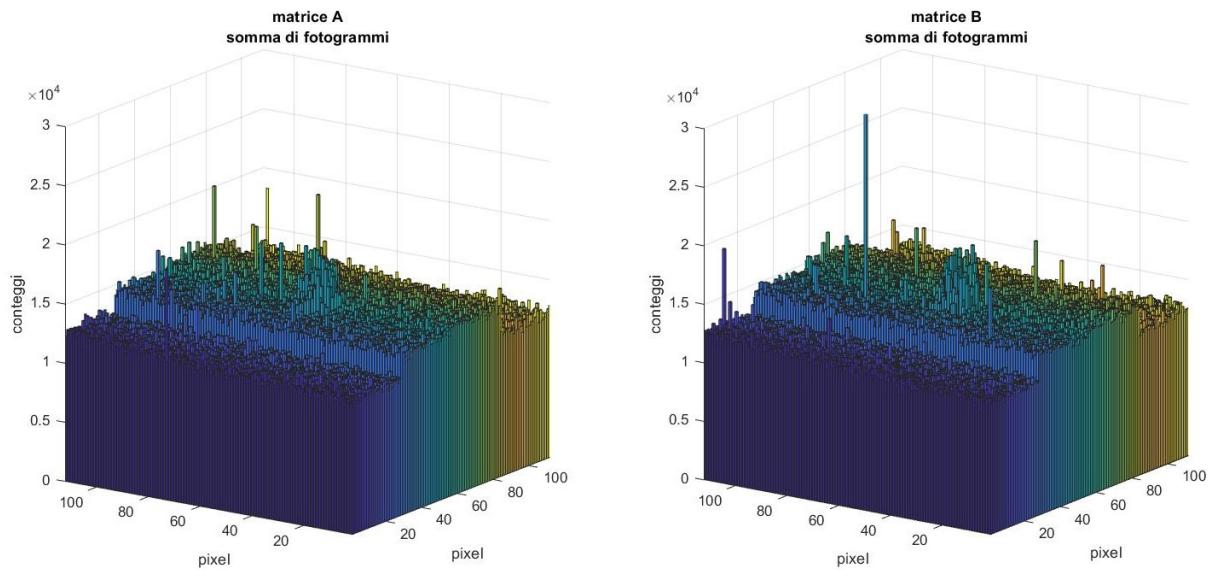


Figura 6: Rappresentazione della somma dei vari fotogrammi, per le matrici di dati A e B.

Dalle immagini così ottenute è possibile distinguere una zona caratterizzata da maggiore intensità: pertanto si è deciso di esaminare le proiezioni per righe e colonne della somma rappresentata.

Di seguito si riportano le proiezioni per righe e colonne delle due matrici-somma del segnale:

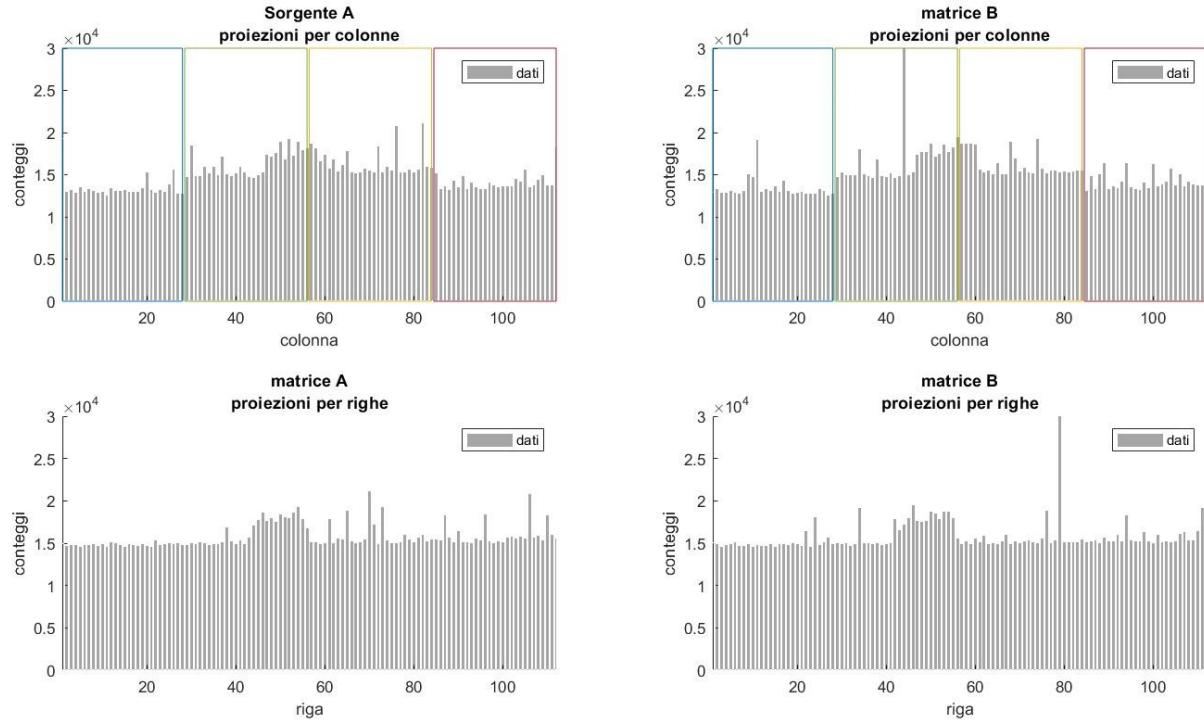


Figura 7: Rappresentazioni delle proiezioni, per righe e colonne, dei segnali in figura (6).

Dalle proiezioni è possibile osservare:

- la sorgente non risulta chiaramente identificabile (specialmente nelle proiezioni per colonne) per via della presenza di piedistalli nei quali risulta immersa la sorgente: pertanto sarà necessario ricorrere a strategie più elaborate che contemplino la presenza di tali piedistalli;
- un'evidente differenza tra il segnale ottenuto nei quadranti centrali e quelli esterni del rivelatore, attribuibile al raggruppamento in sotto-matrici dei pixel che compongono il rivelatore stesso.

2.2 Equalizzazione

La fase di equalizzazione si è quindi concentrata sull'identificazione di valori attribuibili ai piedistalli, osservati nella sezione precedente e da sottrarre ai dati considerati.

Questo procedimento è stato svolto in due *step*: è stato prima calcolato un valore di piedistallo per ciascun quadrante considerato, che è risultato però insufficiente e si è quindi passati all'individuazione di un piedistallo per ciascun pixel considerato.

2.2.1 Sottrazione di piedistallo per quadrante

Come è possibile osservare sia dalla figura (6) sia dalla figura (7) sono presenti quattro quadranti per i quali è possibile ricavare un valore di piedistallo da sottrarre ai dati.

Per i due quadranti più esterni, siccome non risultano particolarmente inficiati dalla presenza della sorgente, il piedistallo è stato calcolato come la media di tutti i segnali registrati da pixel appartenenti al quadrante in questione.

Invece, per i due quadranti centrali che contengono la sorgente, sono state considerate le sottosezioni più esterne (in modo tale da non includere la sorgente): tali zone sono state utilizzate per effettuare una media dei segnali da utilizzare come valore di piedistallo per ciascuno.

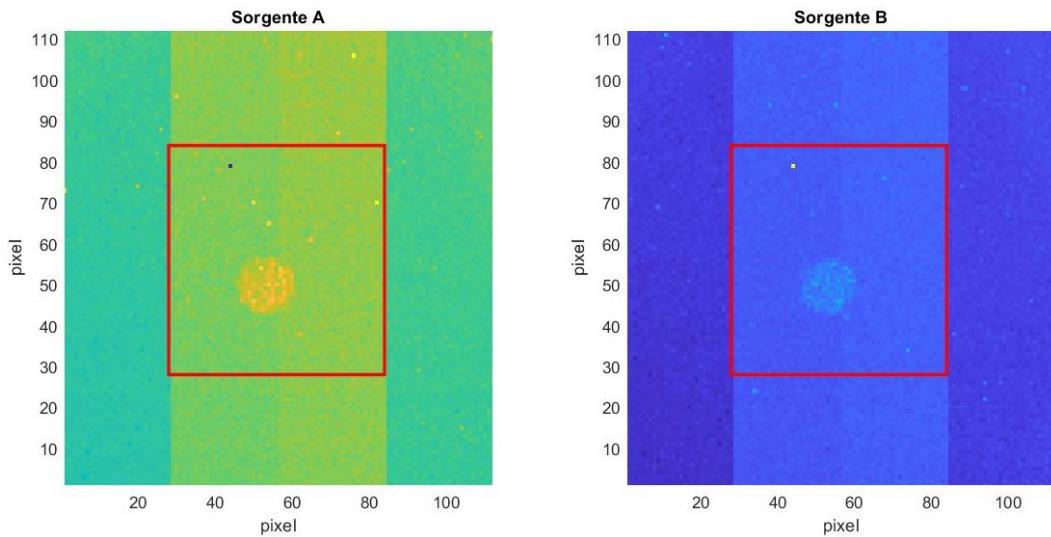


Figura 8: Rappresentazione delle zone escluse dal calcolo del piedistallo (interne al quadrato in rosso), per le matrici di dati A e B.

I valori di piedistallo così ottenuti sono stati utilizzati per compilare una matrice (di piedistalli) da sottrarre a ciascun fotogramma, in modo da ottenere un segnale più vicino a rappresentare la reale energia depositata dall'impatto delle particelle con il rivelatore.

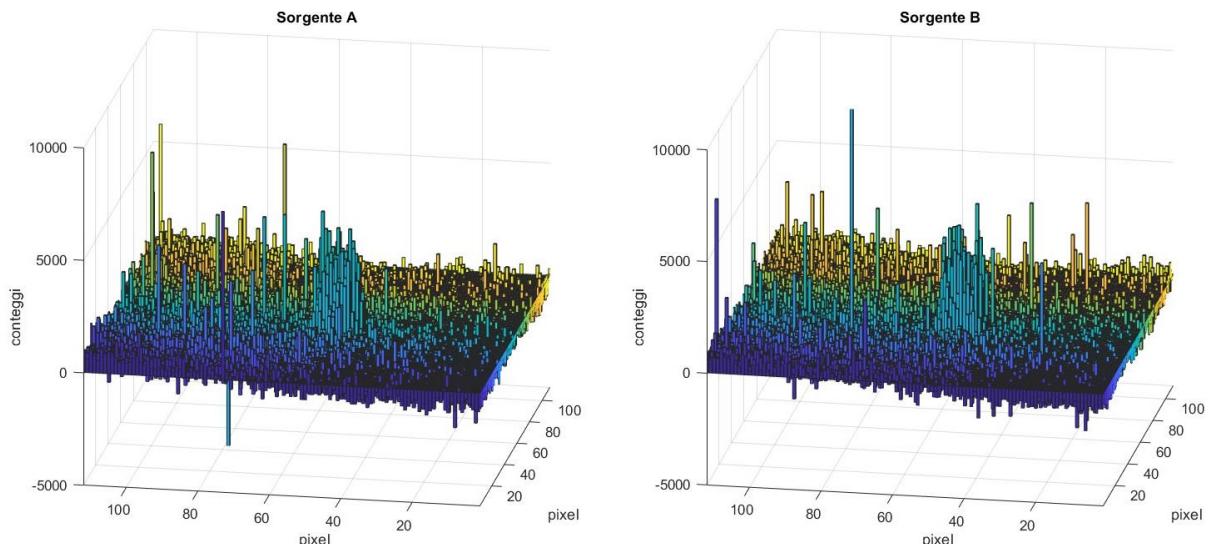


Figura 9: Somma dei *frame* a cui è stato sottratto il piedistallo trovato, per le matrici A e B.

I grafici in figura (9) sono stati ottenuti sommando i fotogrammi precedentemente menzionati e ai quali è stata sottratta la matrice di piedistalli. È possibile osservare che la sorgente non è ancora ben distinguibile per via della presenza di pixel anomali ed una crescita del *plateau* all'aumentare della riga di pixel considerata. Tale comportamento risulta evidente nelle proiezioni per riga in figura (7): se infatti la disomogeneità tra i quadranti osservata in precedenza risulta così corretta, la crescita per righe rimane un problema da risolvere che giustifica la decisione di utilizzare una seconda strategia di equalizzazione.

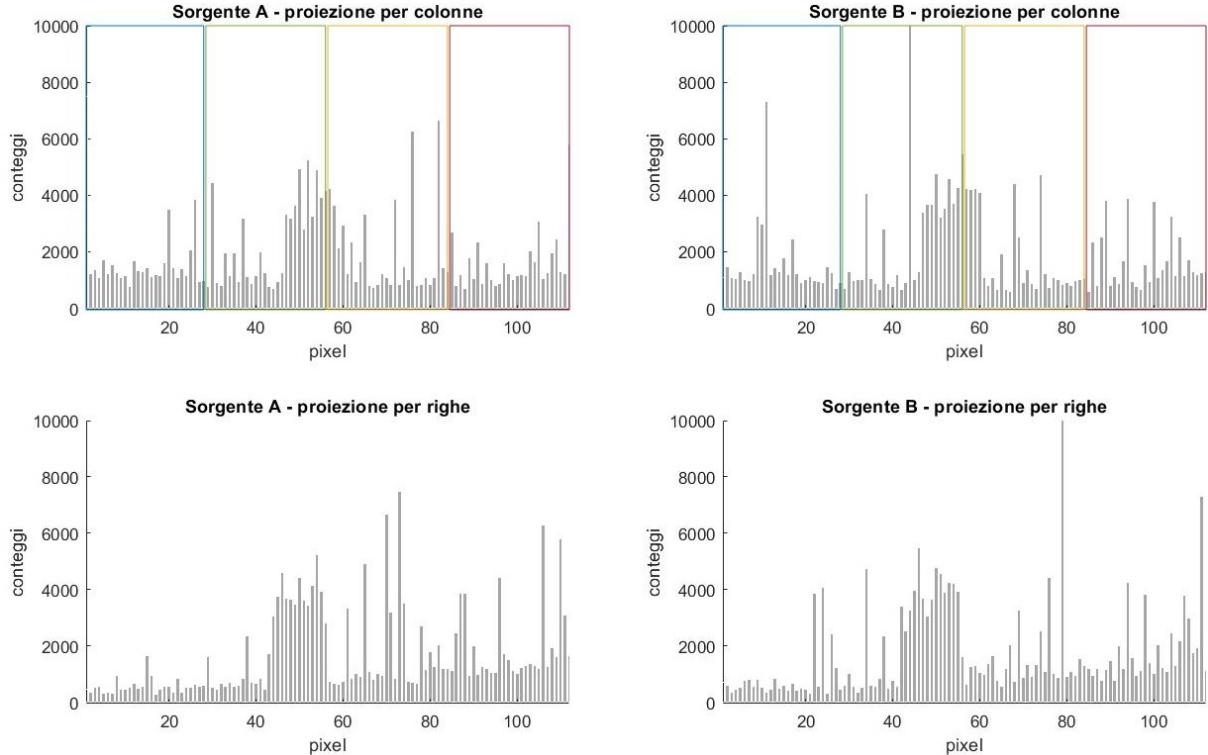


Figura 10: Rappresentazione delle proiezioni, per righe e colonne, dei segnali in figura (9).

2.2.2 Sottrazione di piedistallo per pixel

Per risolvere le anomalie riscontrate nella sezione precedente si è deciso di implementare una strategia basata sull'identificazione di un piedistallo per ciascun pixel (di coordinate (i, j)).

Per fare ciò sono stati considerati la media \bar{x}_{ij} (su tutti i *frame*) dei segnali rivelati dal pixel e il relativo errore sulla media² $\sigma_{\bar{x}_{ij}}$: questi sono stati utilizzati per calcolare una soglia³ $s = \bar{x}_{ij} + 5\sigma_{\bar{x}_{ij}}$, necessaria per identificare (per esclusione, con un livello di confidenza del 99.9999 %) i segnali corrispondenti a registrazioni di urti. Questi infatti non devono essere inclusi nel processo di stima del piedistallo del pixel, il quale viene calcolato come la media dei segnali compatibili con l'assenza di urti, cioè al di sotto di s_{ij} .

Come in precedenza l'analisi della sorgente è stata effettuata analizzando la somma dei fotogrammi a ciascuno dei quali è stata sottratta la matrice contenente i piedistalli per pixel.

²vedi 4.1 Appendice: Media e relativa incertezza.

³La zona di confidenza a 5 sigma è stata delimitata solo superiormente (con +) siccome si stanno utilizzando, come già menzionato, i moduli dei segnali registrati.

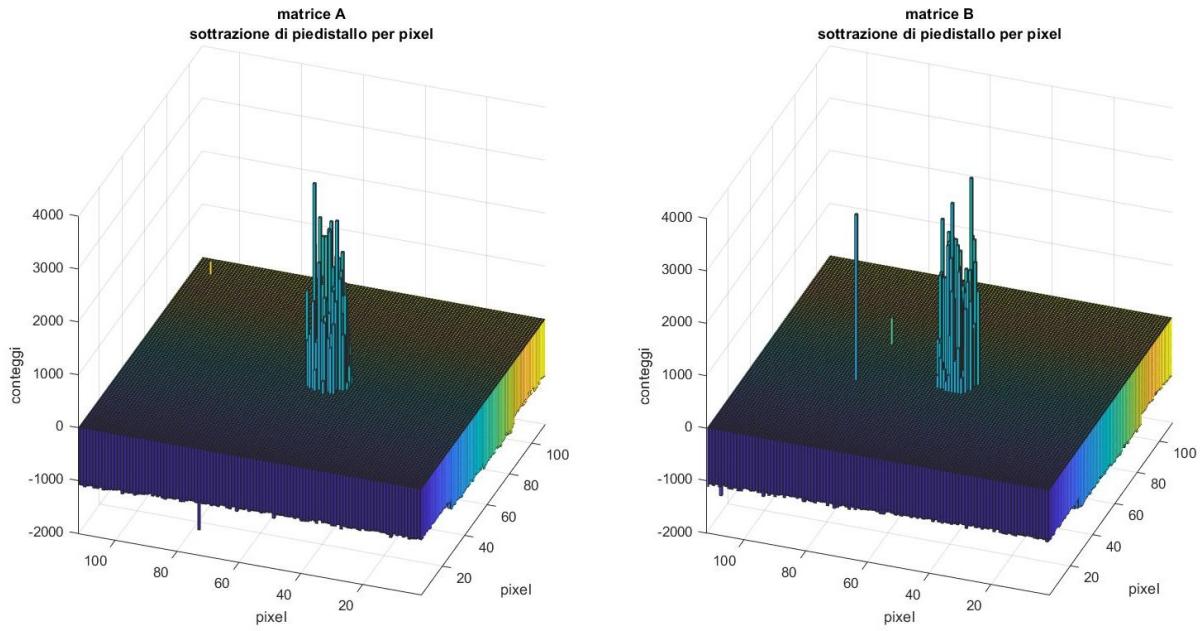


Figura 11: Somma dei *frame* a cui è stato sottratto il piedistallo trovato per ciascun pixel, per le matrici A e B.

La strategia appena descritta ha permesso di ottenere un’immagine più nitida della sorgente: infatti è stato persino possibile determinare la presenza di pixel rotti (uno dei quali particolarmente visibile siccome alto quanto la sorgente) che risultano visibili anche nelle proiezioni per righe e colonne riportate a seguire.

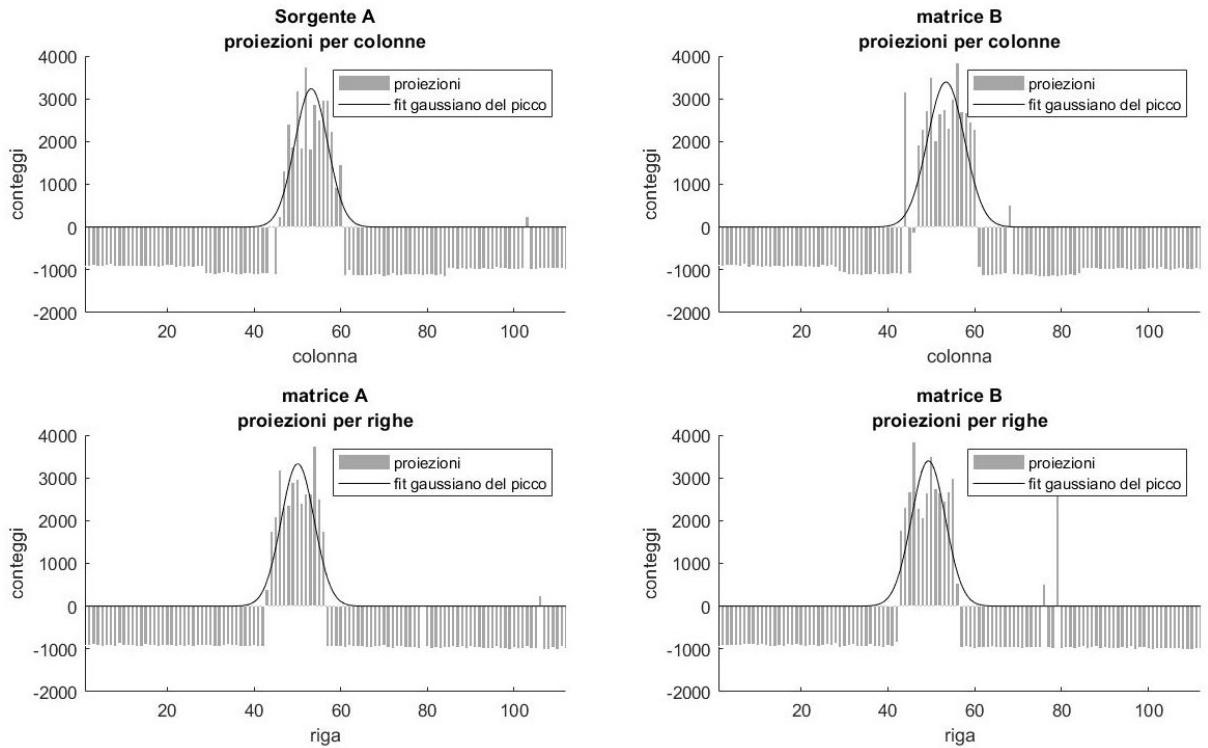


Figura 12: Rappresentazione delle proiezioni, per righe e colonne, dei segnali in figura (11), con *fit gaussiani*.

Come è possibile osservare in figura (12), la strategia utilizzata si è rivelata soddisfacente anche dal punto di vista delle proiezioni, necessarie per ottenere informazioni relative all'energia rilasciata dalla sorgente sul rivelatore e quindi sulla sorgente stessa.

In particolare sono state ricavate le coordinate (x, y) del centro e le dimensioni del picco: le prime sono state stimate come i valori medi dei *fit gaussiani* (limitati ai picchi) delle proiezioni in figura (12). Le dimensioni sono invece state misurate come la larghezza dei medesimi picchi: si indica con a la larghezza del picco nella proiezione per righe e con b la larghezza del picco nella proiezione per colonne.

	A	B
x [pixel]	53.2 ± 0.6	53.4 ± 0.7
y [pixel]	50.1 ± 0.6	49.3 ± 0.6
a [pixel]	14 ± 2	14 ± 2
b [pixel]	14 ± 2	14 ± 2

Tabella 1: Dimensioni e coordinate del centro delle sorgenti.

Si osserva che i valori ottenuti per i due assi a e b , riportati in tabella (1), risultano identici: in questo modo l'ellisse di assi a e b , utilizzata per modellizzare la sorgente, degenera in una circonferenza di diametro $a = b$.

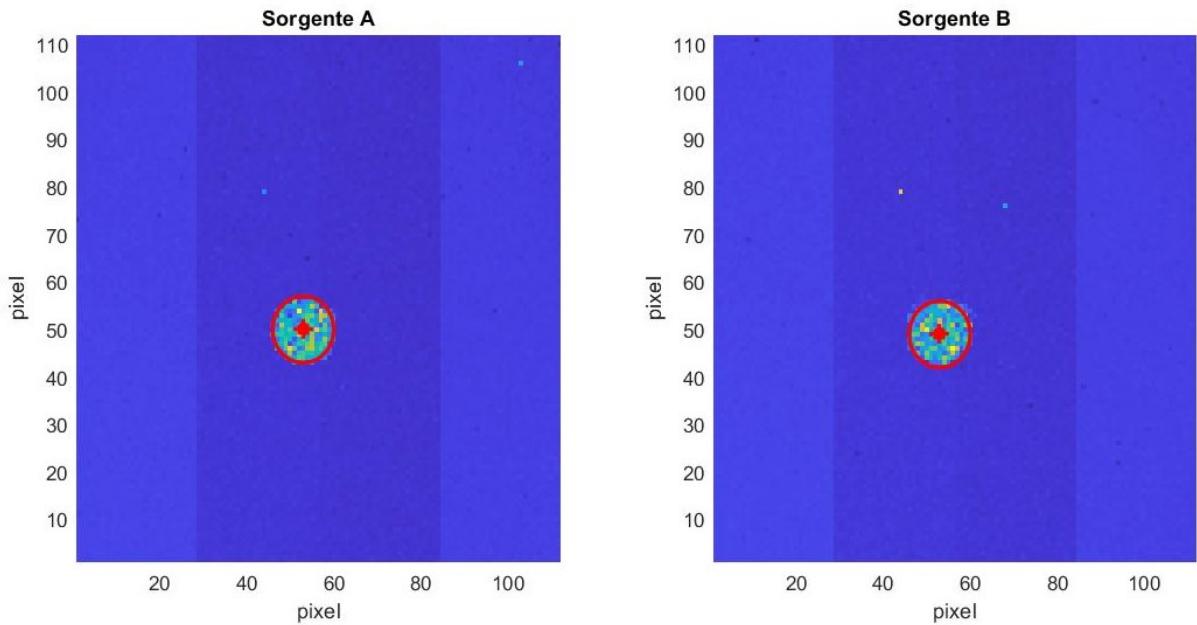


Figura 13: Rappresentazione della forma delle sorgenti, ottenuta con l'equalizzazione.

2.3 Quantum imaging e individuazione dei cluster

Per determinare in maniera piú precisa le dimensioni della sorgente è possibile utilizzare una strategia alternativa a quella descritta in precedenza, che sfrutta anch'essa i valori di piedistallo per pixel calcolati nella sezione precedente.

Confrontando le soglie s_{ij} con i valori di segnale assunti nei vari fotogrammi, è possibile osservare la presenza di due tipologie di pixel (visualizzabili in figura (14)):

- alcuni sono caratterizzati dalla presenza di *outliers*, ovvero segnali al di sopra della soglia s_{ij} a 5σ : sapendo quindi che l'ipotesi nulla iniziale consiste nella compatibilità con le oscillazioni stocastiche (attribuibili al segnale elettrico), si deduce (con un livello di confidenza pari a 99.9999 %) che a tali *outliers* corrispondono effettivi urti di particelle.
É quindi possibile definire i pixel in questione *accesi* (i quali si trovano prevalentemente in corrispondenza della sorgente);
- i restanti sono caratterizzati dall'assenza di *outliers* siccome tutti i dati rientrano nella fascia di compatibilità a 5σ . I segnali così fatti risultano quindi compatibili con le oscillazioni stocastiche attribuibili al segnale elettrico ed è quindi possibile definire i pixel in questione *spenti* (che si trovano al di fuori della zona d'interesse della sorgente).

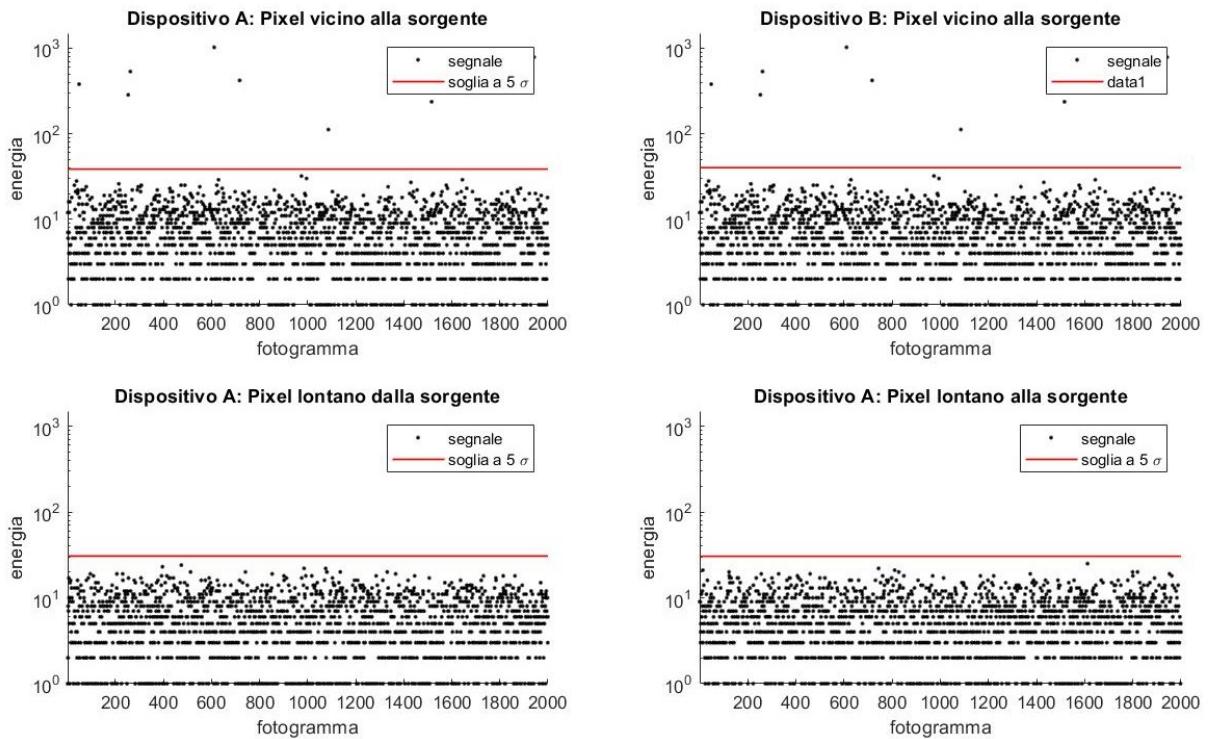


Figura 14: Esempi di segnale di un pixel acceso (vicino alla sorgente) e di uno spento (lontano dalla sorgente), per le matrici di dati A e B.

Viste tali considerazioni sui pixel è stato possibile trascurare i contributi di segnale dati dai pixel spenti (in quanto attribuibili alle fluttuazioni stocastiche del segnale elettrico) per concentrarsi unicamente sul comportamento di quelli accesi. In particolare, come è possibile notare in figura (14), anche il segnale di un pixel acceso risulta per la maggior parte dei fotogrammi al di sotto della soglia di compatibilità a 5σ . Quindi per ottenere informazioni sugli urti e sulla sorgente è sufficiente analizzare i *frame* in cui il pixel considerato è acceso.

Sarebbe quindi possibile, in prima analisi, stimare la forma della sorgente semplicemente contando e rappresentando il numero di fotogrammi in cui un dato pixel risulta acceso (cioè subisce un urto). Avendo però una matrice iniziale di soli 4000 *frame*, tale approccio risulta persino meno soddisfacente delle strategie precedenti, poiché i risultati ottenuti sono caratterizzati da un ridotto rapporto segnale-rumore, e viene quindi tralasciato.

La strategia illustrata di seguito permette invece di ottenere un ottimo rapporto segnale-rumore e quindi di identificare chiaramente la sorgente, siccome fa uso dei valori di energia corrispondenti agli urti (molto maggiori dei numeri di fotogrammi utilizzati nel metodo tralasciato).

Anche in questo caso è stata sfruttata la caratterizzazione dei pixel in accesi (1) o spenti (0) in ogni fotogramma: essa ha permesso di applicare un metodo di individuazione dei punti di impatto delle particelle basato sulla possibilità che l'impatto di una singola particella α si manifesti come registrazione del segnale in più pixel adiacenti, che risultano quindi contemporaneamente accesi formando dei *cluster*. Quindi l'energia depositata dalla particella durante il singolo urto equivale alla somma dei segnali registrati da tutti i pixel che costituiscono i *cluster*; inoltre è possibile approssimare il punto di impatto di tale particella con il baricentro (media pesata con le energie) del *cluster*.

Risulta essenziale il processo di individuazione dei *cluster* che in questo caso è stato svolto dalla funzione `PacMan` (basata sull'utilizzo della funzione iterativa `PacManFunction`), che richiede in *input* una matrice di booleani che indica, fotogramma per fotogramma, quali pixel risultano accesi e quali spenti. In *output* invece restituisce un *cell-array* contenente il numero di pixel di ciascun *cluster* e le corrispondenti coppie di coordinate (compreso il fotogramma di appartenenza): queste ultime sono infatti individuate analizzando, a partire da un primo pixel acceso, le quattro possibili direzioni che si generano ogni volta che viene rilevato un pixel adiacente acceso.

Gli output della funzione `PacMan` vengono quindi sfruttati per trovare le coordinate dei baricentri degli urti e le relative energie: queste ultime vengono sommate (su tutti i fotogrammi) in modo tale da ottenere finalmente l'immagine della sorgente, riportata di seguito.

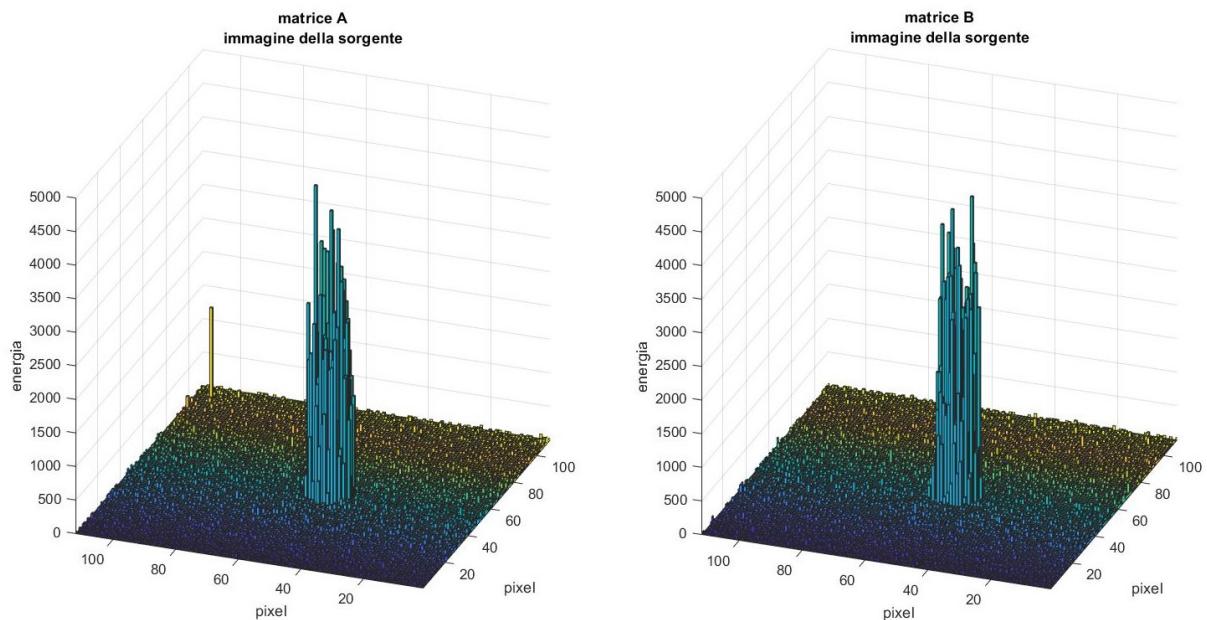


Figura 15: Somma delle energie dei *cluster* ottenuti nei diversi *frame*, per le matrici A e B.

La figura appena riportata risulta effettivamente avere un soddisfacente rapporto segnale-rumore, che si suppone possa portare ad una corretta individuazione della sorgente. Pertanto sono state analizzate le proiezioni per righe e colonne dei risultati ottenuti in figura (15).

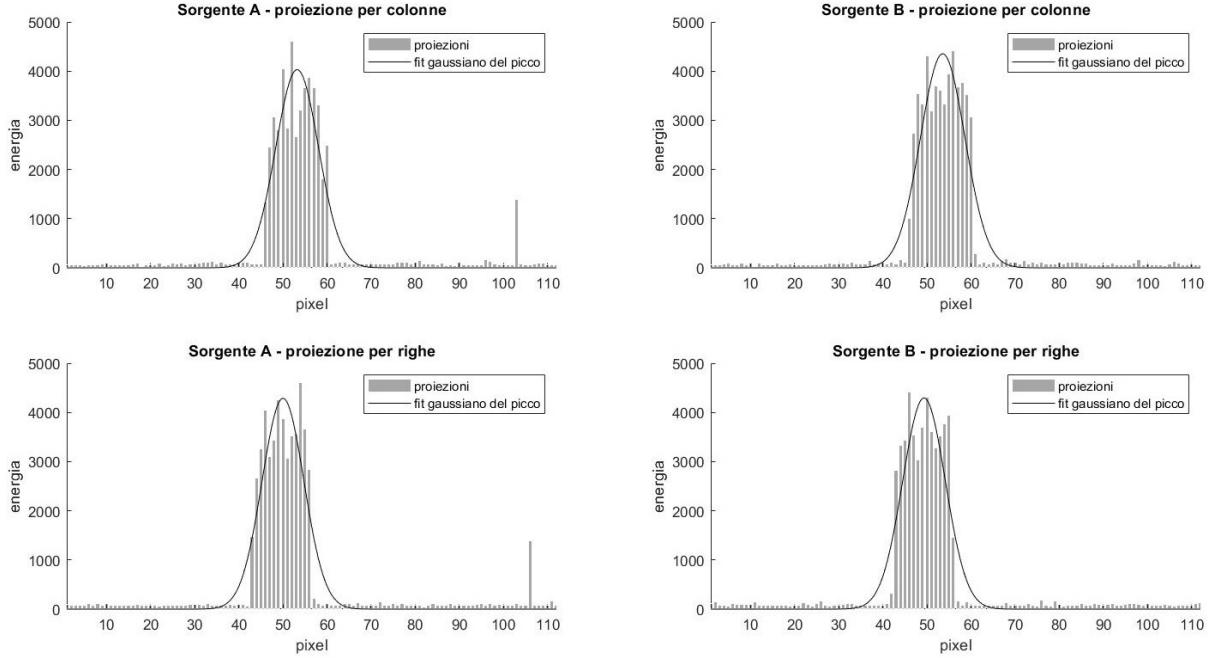


Figura 16: Rappresentazione delle proiezioni, per righe e colonne, dei segnali in figura (15), con *fit* gaussiani.

Come è possibile osservare in figura (16), la strategia utilizzata si è rivelata soddisfacente anche dal punto di vista delle proiezioni, necessarie per ottenere informazioni relative all’energia rilasciata dalla sorgente sul rivelatore e quindi sulla sorgente stessa.

In particolare, anche in questo caso, sono state ricavate le coordinate (x, y) del centro e le dimensioni del picco: in modo analogo al precedente, le prime sono state stimate come i valori medi dei *fit* gaussiani (limitati ai picchi) delle proiezioni in figura (16). Le dimensioni sono invece state misurate come la larghezza dei medesimi picchi: anche in questo caso si indica con a la larghezza del picco nella proiezione per righe e con b la larghezza del picco nella proiezione per colonne.

	A	B
x [pixel]	53.2 ± 0.2	53.6 ± 0.2
y [pixel]	50.0 ± 0.2	49.3 ± 0.2
a [pixel]	14 ± 2	14 ± 2
b [pixel]	14 ± 2	14 ± 2

Tabella 2: Dimensioni e coordinate del centro delle sorgenti.

Si osserva nuovamente che i valori ottenuti per i due assi a e b , riportati in tabella (2), risultano identici e l’ellisse di assi a e b , utilizzata per modellizzare la sorgente, degenera in una circonferenza di diametro $a = b$, visualizzabile in figura (17).

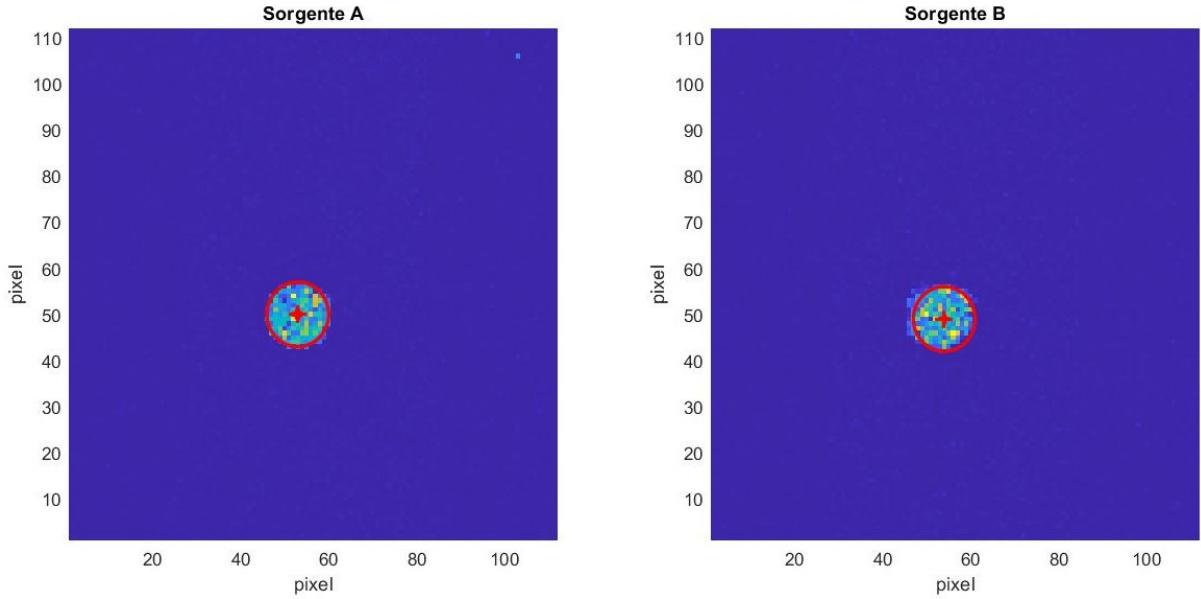


Figura 17: Rappresentazione della forma delle sorgenti, ottenuta tramite l’individuazione dei *cluster*.

3 Conclusioni

Le due matrici di dati (A e B) sono state sottoposte alle due strategie di analisi, l’equalizzazione e l’individuazione di *cluster*, per individuare la forma della sorgente (cioè del collimatore) a partire dall’analisi delle proiezioni. I due metodi implementati hanno restituito risultati soddisfacenti, compatibili tra di loro e caratterizzati da incertezze dello stesso ordine di grandezza (osservabile dalle tabelle (1) e (2)).

Come è possibile visualizzare in figura (15), il metodo di individuazione dei *cluster* ha permesso però di ottenere un migliore rapporto segnale-rumore anche per via dell’eliminazione dei piedistalli conseguente all’individuazione dei pixel accesi/spentti per ciascun *frame*.

Pertanto, le dimensioni trovate con questo metodo risultano lievemente⁴ più precise e saranno quindi utilizzate per convertire le misure ottenute (in pixel) in unità di lunghezza.

È nota infatti la dimensione di ciascun pixel, pari a $d = 153 \mu\text{m}$: supponendo che le fughe tra i vari pixel siano trascurabili rispetto alla dimensione dei pixel stessi, risulta possibile determinare la dimensione della sorgente, moltiplicando i dati in tabella (2) per la dimensione d e ottenendo i seguenti risultati:

	A	B
x [mm]	8.14 ± 0.03	8.20 ± 0.03
y [mm]	7.65 ± 0.03	7.54 ± 0.03
a [mm]	2.1 ± 0.3	2.1 ± 0.3
b [mm]	2.1 ± 0.3	2.1 ± 0.3

Tabella 3: Dimensioni e coordinate del centro delle sorgenti convertite in unità di lunghezza.

⁴La precisione non risulta drasticamente migliorata per via del fatto che la maggior parte dei *cluster* trovati risulta composta da un pixel e l’algoritmo non fa così tanta differenza.

4 Appendice

4.1 Media e relativa incertezza

Sia x_i un set di N dati. Si definisce *valore medio*:

$$\bar{x} = \frac{1}{N} \cdot \sum_{i=1}^N x_i \quad (2)$$

Inoltre, è possibile attribuire a questo valore un'incertezza pari a:

$$\sigma_{\bar{x}} = \frac{\text{std}(x_i)}{\sqrt{N}} \quad (3)$$

Dove $\text{std}(x_i)$ è la *deviazione standard* che può essere calcolata tramite la seguente formula:

$$\text{std}(x_i) = \sqrt{\frac{1}{N-1} \cdot \sum_{i=1}^N |x_i - \bar{x}|^2} \quad (4)$$

4.2 Codice per individuazione dei *cluster*

A seguire si riporta il codice MATLAB per la funzione PacMan di identificazione dei *cluster*, la quale ricorre alla funzione iterativa PacManFunction. L'efficienza della funzione è stata testata facendole analizzare una matrice tridimensionale ($5000 \times 10 \times 10$) di booleani scelti randomicamente e ha dimostrato un'efficienza del 99.99%.

```

1 function [Matrix2d,InfoCluster,pix] = PacManFunction(Matrix2d,InfoCluster,pix)
2
3 % Funzione che, dati:
4 % - la matrice di booleani in questione (che serve come traccia per la ricerca degli ulteriori pixel del cluster);
5 % - una matrice (per ogni cluster) che ospita (rispettivamente nella prima e nella seconda colonna) le righe e le
6 %   colonne dei pixel che fanno parte in questione del cluster, mentre nella terza è presente la direzione in cui
7 %   è possibile spostarsi dal primo pixel;
8 % - il valore del booleano assunto dal pixel
9 % restituisce gli stessi input ma aggiornati al pixel esaminato.
10
11 % Inizializziamo la dimensione dei frame, necessari per il funzionamento della seguente funzione iterativa:
12
13 dimMat=size(Matrix2d,2);
14
15 % Nel caso in cui il cluster sia formato da un unico pixel si ha:
16
17 if pix==0
18     return
19
20 % invece, nel caso in cui si possano trovare ulteriori pixel:
21
22 else
23     indRiga=InfoCluster(pix,1);      % indice riga
24     indColonna=InfoCluster(pix,2);    % indice colonna
25
26 % Se abbiamo già esaminato tutte le possibili direzioni, significa che il cluster è stato esaminato completamente:
27
28 if InfoCluster(pix,3)==4
29     [Matrix2d,InfoCluster,pix]=PacManFunction(Matrix2d,InfoCluster,pix-1);
30     return
31
32 % In caso contrario, è necessario modificare l'indice di direzione:
33 % - se la direzione possibile è la destra incrementiamo l'indice di colonna in modo da spostarci:
34
35 elseif InfoCluster(pix,3)==0
36     indColonna=indColonna+1;

```

```

3/
38 % - se la direzione possibile è sotto incrementiamo l'indice di riga in modo da spostarci:
39
40     elseif InfoCluster(pix,3)==1 % sotto
41         indRiga=indRiga+1;
42
43 % - se la direzione possibile è la sinistra diminuiamo l'indice di colonna in modo da spostarci:
44
45     elseif InfoCluster(pix,3)==2 % sinistra
46         indColonna=indColonna-1;
47
48 % - se la direzione possibile è sopra diminuiamo l'indice di riga in modo da spostarci:
49
50     elseif InfoCluster(pix,3)==3 % sopra
51         indRiga=indRiga-1;
52     end
53
54 % Una volta fatto ciò, è necessario evitare che l'algoritmo indaghi oltre le dimensioni della matrice:
55
56     if indRiga>dimMat || indRiga<=0 || indColonna>dimMat || indColonna<=0
57
58 % facendo in modo di incrementare l'indice di direzione quando l'algoritmo è sul bordo o fuoriesce dal bordo:
59
60         InfoCluster(pix,3)=InfoCluster(pix,3)+1;
61
62 % e richiamando la funzione con questa modifica per fare in modo che avanzi correttamente:
63
64     [Matrix2d,InfoCluster,pix] = PacManFunction(Matrix2d,InfoCluster,pix);
65     return
66
67 % Se tutto va bene con il bordo, allora è possibile finalmente analizzare il pixel successivo:
68
69     else
70
71 % se il nuovo pixel è acceso
72
73     if Matrix2d(indRiga,indColonna)==1
74
75 % sostituiamo -1 a 1 (in modo poi da poterlo distinguere dai pixel originariamente spenti, ma soddisfando
76 % sempre la condizione iniziale del while):
77
78         Matrix2d(indRiga,indColonna)=-1;
79
80 % settiamo il nuovo indice di direzione a 0
81
82         InfoCluster=[InfoCluster;[indRiga,indColonna,0]];
83
84 % e anche il numero di pixel individuati in ciascun cluster
85
86         pix=length(InfoCluster(:,1));
87
88 % e richiamiamo nuovamente la funzione:
89
90         [Matrix2d,InfoCluster,pix] = PacManFunction(Matrix2d,InfoCluster,pix);
91         return
92
93 % Nel caso in cui il pixel sia spento o già esaminato:
94
95     else
96
97 % incrementiamo semplicemente l'indice di direzione:
98
99         InfoCluster(pix,3)=InfoCluster(pix,3)+1;
100
101 % e richiamiamo nuovamente la funzione:
102
103         [Matrix2d,InfoCluster,pix] = PacManFunction(Matrix2d,InfoCluster,pix);
104         return
105     end
106
107     end
108
109 end
110
```

```

1 function [all_info_cluster,errori] = PacMan(bool_mat)
2
3 % Questa funzione prende in input la matrice di booleani corrispondente ai pixel accesi (indicati con 1) e ai
4 % pixel spenti (indicati con 0) e ne ricava le informazioni utili sui cluster presenti, che raccoglie nell'output
5 % all_info_cluster. Ulteriore output sono gli errori di esecuzione, necessari per verificare il corretto
6 % funzionamento della funzione.
7
8 % Per garantire la leggibilità, definiamo il numero di frame che compongono la matrice-3D:
9
10 n_frame=size(bool_mat,3);
11
12 % e inizializziamo:
13 % - un cell-array contenente le info sui cluster:
14 all_info_cluster=cell(1,3);
15
16 % - il numero di errori dell'algoritmo:
17 n_error=zeros(n_frame,1);
18
19 % cicliamo per esaminare ciascun frame della matrice-3d:
20
21 for frame=1:n_frame
22
23 % Inizializziamo:
24 % - un cell-array info_cluster (che verrà chiamato AllInfoCluster una volta che il frame sarà totalmente
25 % esaminato) in cui sono salvate:
26 % * nella prima colonna, le coppie di coordinate dei pixel che compongono il cluster;
27 % * nella seconda colonna, il numero di pixel che compongono il cluster;
28 % * nella terza colonna, il frame in cui è stato individuato il cluster;
29
30 info_cluster=cell(1,3);
31
32 % - una matrice, inizialmente uguale al frame di booleani che verrà modificata per far progredire l'algoritmo
33 % tramite un ciclo while:
34
35 mat_2d=bool_mat(:,:,frame);
36 while any(mat_2d>0,'all')
37
38 % Individuiamo ora la posizione del primo elemento positivo nella matrice:
39
40 [colonna,riga]=find(mat_2d'==1);
41 riga=riga(1);
42 colonna=colonna(1);
43
44 % lo modifichiamo subito a -1, in modo tale da evitare che venga contato più volte:
45
46 mat_2d(riga,colonna)=-1;
47
48 % Utilizziamo il primo pixel come punto di partenza per la ricerca degli altri pixel (se esistono) del cluster
49 % tramite la funzione PacManFunction:
50
51 [mat_2d,info_cluster{end+1,1},pixel] = PacManFunction(mat_2d,[riga,colonna,0],1);
52
53 % Una volta esaminato il cluster, eliminiamo l'ultima colonna, contenente l'indice di direzione che risulta utile
54 % solamente in PacManFunction e salviamo quindi la matrice rimanente, contenente le coordinate dei pixel
55 % che compongono il cluster nella prima colonna; salviamo anche, nella seconda colonna, il numero di pixel che
56 % compongono il cluster e nella terza colonna il frame/matrice in cui il cluster è stato trovato:
57
58 info_cluster{end,1}=info_cluster{end,1}(:,1:2);
59 info_cluster{end,2}=size(info_cluster{end,1}(:,1:2),1);
60 info_cluster{end,3}=frame;
61
62 end
63
64
65 % Una volta individuati tutti i cluster presenti in un frame, appendiamo i loro dettagli al cell-array per l'intera
66 % matrice-3d (in output alla funzione), eliminando la prima riga (vuota, per l'inizializzazione di info_cluster):
67
68 all_info_cluster=[all_info_cluster(:,:,1) ; info_cluster(2:end,:)];
69
70 % In particolare, nel primo frame entra in gioco anche l'inizializzazione di all_info_cluster:
71
72 if frame==1
73 all_info_cluster=all_info_cluster(2:end,:);
74 end
75
76 % Passiamo ora alla parte di verifica dell'attendibilità del nostro algoritmo: sfruttiamo in particolare la
77 % modalità di modifica della matrice impiegata, che si basa sulla sostituzione di 1 con -1 (una volta che i
78 % pixel sono stati tenuti conto) scommendo la matrice originale a quella temporanea modificata.
79 % Se il processo avviene correttamente si ottiene una matrice di valori nulli:
80
81 mat_sum=bool_mat(:,:,:,frame)+mat_2d;
82
83 % per verificare la presenza di incongruenze basta sommare tutti i valori presenti nella matrice (teoricamente
84 % di valori nulli) appena trovata:
85
86 n_error(frame)=(sum(abs(mat_sum),'all')>0);
87 end
88
89 % Restituiamo, quindi, come secondo output, il numero di frame della matrice-3d contenenti incongruenze:
90
91 errori=sum(n_error);
92
93 end
94
```

4.3 Analisi supplementare del *set* di dati Stuzzicadenti

Infine, applicando le stesse due strategie implementate nell'analisi precedente, è stato possibile analizzare anche un secondo *set* di dati forniti, acquisiti in condizioni differenti dai precedenti (che si presuppone coinvolgano una sorgente vicina al collimatore a tal punto da evidenziarne la presenza sotto forma di taglio nelle immagini).

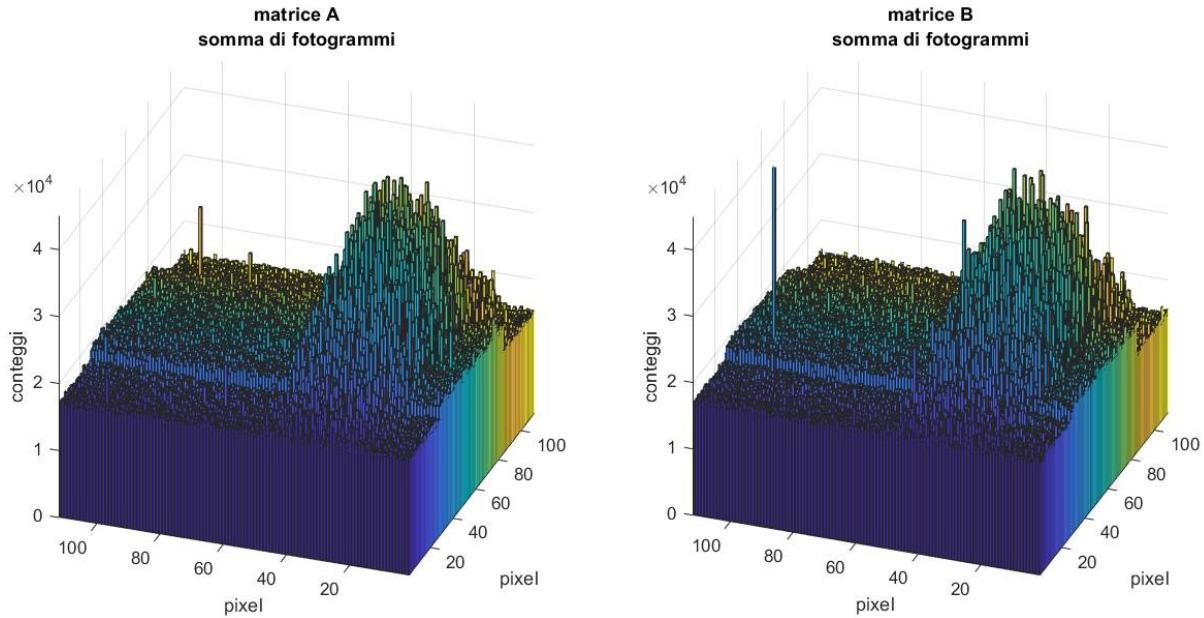


Figura 18: Rappresentazione della somma dei vari fotogrammi, per le matrici di dati A e B.

In questo caso la zona d'interesse risulta essere l'intera metà inferiore della matrice: per evitare che inficiasse il calcolo del piedistallo per quadrante è stata quindi esclusa.

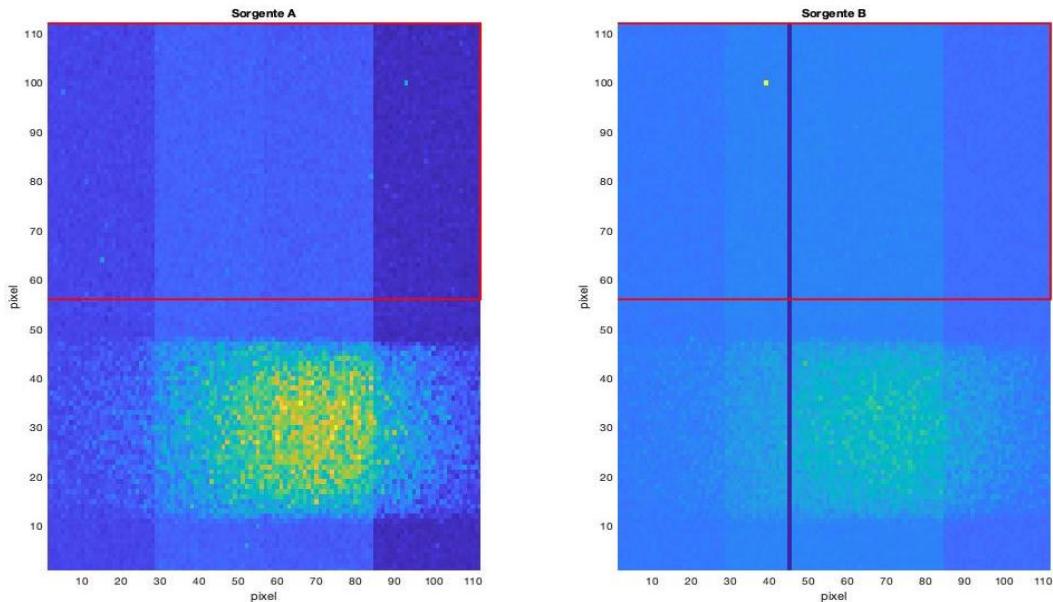


Figura 19: Rappresentazione delle zone incluse nel calcolo del piedistallo per quadrante (interne al quadrato in rosso), per le matrici di dati A e B.

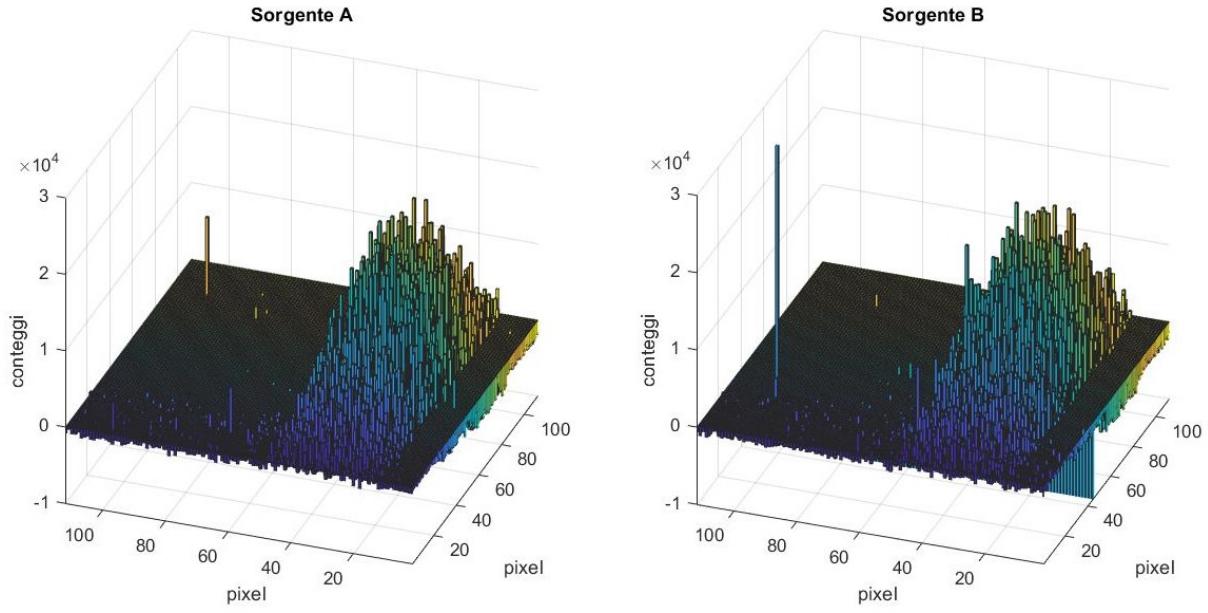


Figura 20: Somma dei *frame* a cui è stato sottratto il piedistallo trovato per quadrante, per le matrici A e B.

Come nel caso precedente è possibile notare anomalie nella matrice B: sono infatti presenti un pixel rotto (avente un segnale maggiore persino di quello della sorgente) e un'intera colonna di pixel danneggiati (a cui corrispondono valori estremamente negativi). La questione è stata risolta individuando un piedistallo per ciascun pixel.

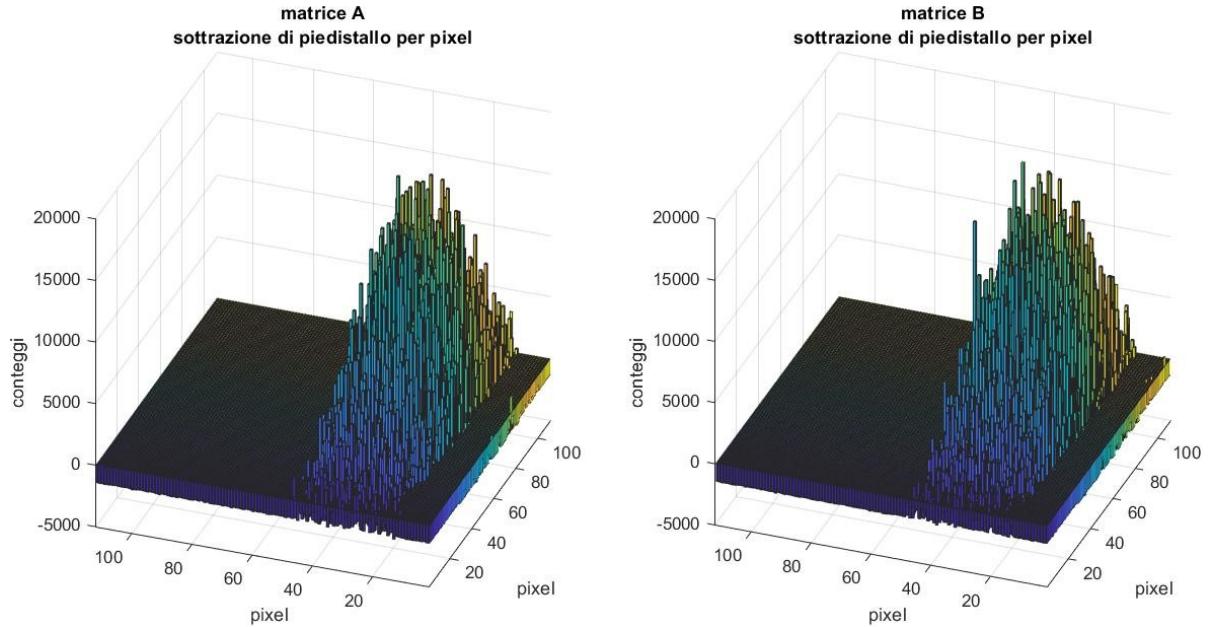


Figura 21: Somma dei *frame* a cui è stato sottratto il piedistallo trovato per ciascun pixel, per le matrici A e B.

La sorgente non risulta ancora abbastanza definita, di conseguenza si è passati direttamente alla seconda strategia senza stimare le sue dimensioni e il centro.

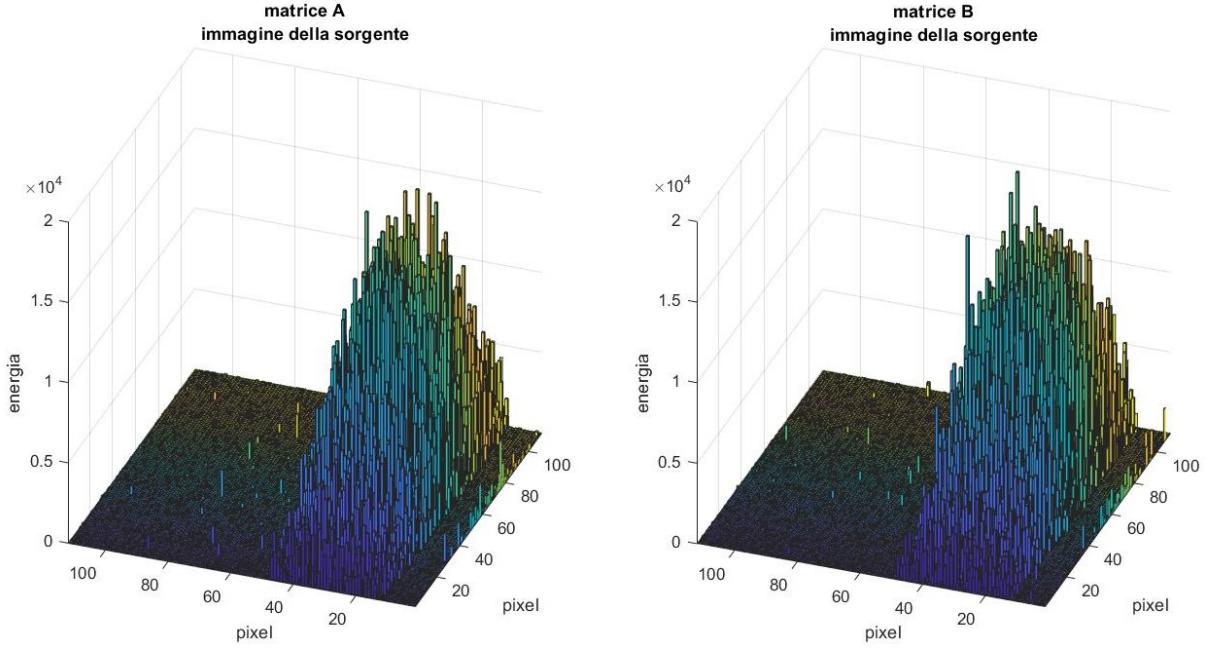


Figura 22: Somma delle energie dei *cluster* ottenuti nei diversi *frame*, per le matrici A e B.

Pur avendo implementato la strategia basata sull'individuazione dei *cluster*, il profilo spaziale della sorgente risulta sempre poco distinguibile. Infatti, anche dalle proiezioni non è stato possibile ricavare le dimensioni della sorgente ma solamente la posizione del suo centro:

$$C_A = (69.3 \pm 0.6, 29.7 \pm 0.4) \text{ pixel} \quad C_B = (69.5 \pm 0.7, 29.7 \pm 0.4) \text{ pixel} \quad (5)$$

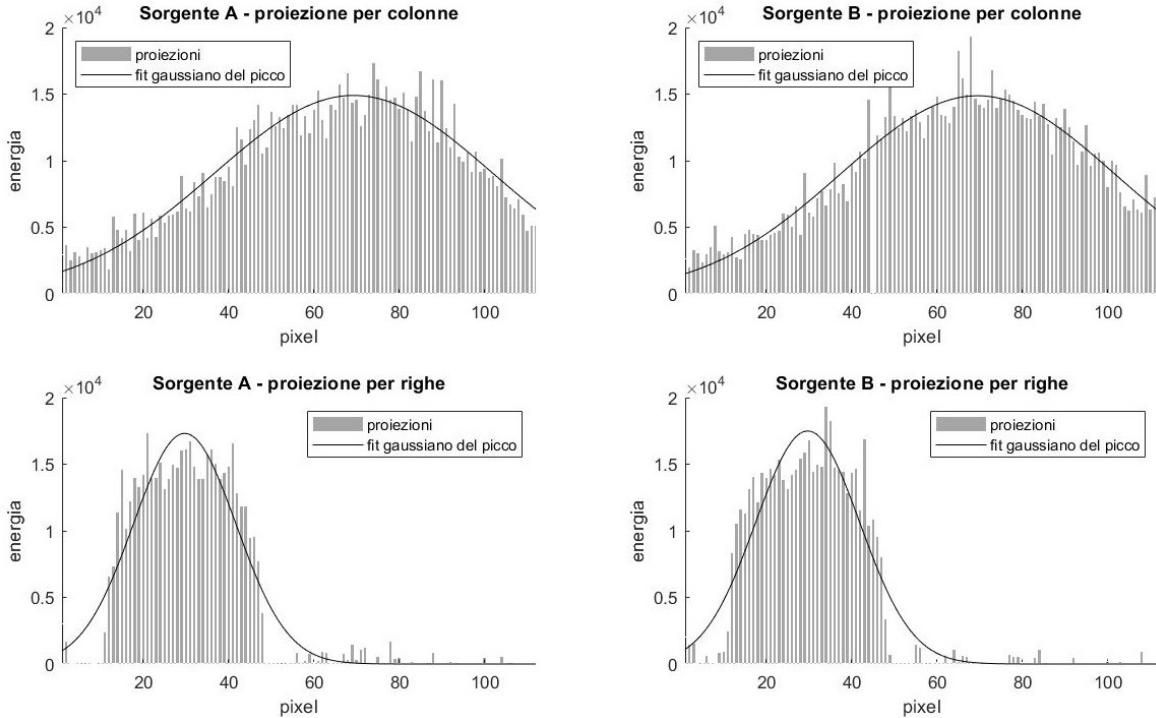


Figura 23: Rappresentazione delle proiezioni, per righe e colonne, dei segnali in figura (22), con *fit gaussiani* utilizzati per stimare le coordinate del centro.

5 Fonti

1. John R. Taylor, *Introduzione all'analisi degli errori* , 1999
2. Maria Garetto, *Statistica: lezioni ed esercizi* , 2002
3. A. Vari, *Scheda del nuclide americio 241*, Alma Mater Studiorum Università di Bologna, 2007