

Prot. n. 817 del 20 dicembre 2001

Anno 2001

*IFAM Rapporto Interno*

N. 1/122001

SVILUPPO DEL SOFTWARE DI  
RICONOSCIMENTO DI UN CAMPO STELLARE

Marco Galimberti<sup>†</sup>, Leonida A. Gizzi<sup>†</sup>, Alessandro Barbini<sup>†</sup>  
Guido Di Cocco<sup>^</sup>, Angelo Basili<sup>^</sup>, Turi Franceschini<sup>^</sup>, Gianni Landini<sup>^</sup>, Stefano Silvestri<sup>^</sup>

<sup>†</sup> *Laboratorio per l'Irraggiamento con Laser Intensi, IFAM, Area della Ricerca  
CNR, Via G. Moruzzi 1, 56124 Pisa, Italy*  
<sup>^</sup> *TESRE, CNR, via P. Gobetti 101, 40129 Bologna, Italy*

# 1 Introduzione

Grazie ai recentissimi sviluppi dell' "ottica" X e  $\gamma$ , per l'astronomia delle alte energie si aprono nuove prospettive. Infatti sono attualmente in fase di realizzazione[2] veri e propri telescopi X e  $\gamma$  dotati di un potere risolutivo ben inferiore al minuto d'arco. Esperimenti da pallone stratosferico sono già in corsa o in fase di realizzazione.

Uno dei principali problemi di questi esperimenti è legato alla precisione di puntamento della piattaforma. Metodi basati sul campo magnetico terrestre, magnetometri, o onde radio, GPS, sono in grado di risolvere la direzione di puntamento con circa un decimo di grado di risoluzione. Uno strumento utilizzabile per migliorare questa precisione, fino a qualche secondo d'arco, richiede l'impiego di uno star-sensor, ovvero di un sensore che si basa sull'acquisizione di un'immagine del campo stellare per poi riconoscerne le stelle[1].

In questo scenario rientra il progetto HIPEG che prevede la realizzazione di una gondola da palloni stratosferici per telescopi X/g. Nel progetto attuale la traiettoria di puntamento finale dell'esperimento sarà ricostruita con precisione di 15 arcsec (goal: 10 arcsec) grazie all'impiego di un sensore stellare/solare basato su un CCD raffreddato, ad alta dinamica (12 bit). Nella sua configurazione finale il sensore stellare fornirà informazioni di puntamento con aggiornamento ogni secondo sia durante il giorno che durante la notte.

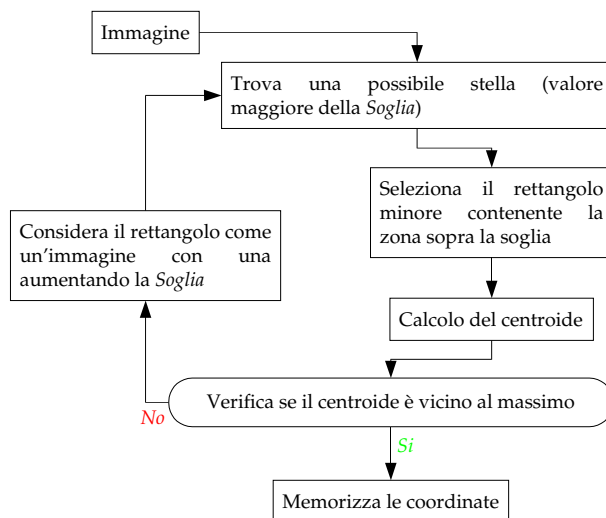
Vengono qui riportati lo sviluppo e i test di un programma di riconoscimento di campo stellare, basato sul catalogo di stelle Hipparcos[3].

Il programma si basa su tre algoritmi: localizzazione delle stelle all'interno dell'immagine acquisita, riconoscimento delle stelle tramite un catalogo esterno e inseguimento del puntamento.

## 2 Localizzazione delle stelle

Primo problema di un riconoscimento di campo stellare consiste nel localizzare le stelle all'interno dell'immagine, selezionando il numero maggiore possibile di stelle ed evitando, o almeno riducendo al minimo, gli artefatti (rumore elettronico, ecc.).

Lo schema dell'algoritmo da noi implementato è mostrato in fig. 1. Partendo dall'immagine, se ne calcola l'istogramma dei valori e se ne trova il massimo. Supponendo che questo rappresenti il livello della luce diffusa dell'immagine, si somma ad esso un valore iniziale in modo da fissare una soglia  $\mathcal{S}$ . Successivamente si scorrono tutti i pixel dell'immagine fino a che se ne trova uno sopra questa soglia, cioè una possibile stella. Si trova il rettangolo più piccolo contenente tutti i pixel maggiori di  $\mathcal{S}$  nell'intorno del primo pixel. Questo rettangolo viene "staccato" e cancellato dall'immagine per poi calcolarne il centroide e la posizione del valore massimo. Se questi due coincidono entro un pixel, la porzione di



**Figura 1:** Schema dell’algoritmo di localizzazione delle stelle nell’immagine.

immagine viene catalogata come stella e se ne salva il centroide e l’integrale del rettangolo, altrimenti si considera il rettangolo come se fosse un’immagine nuova con una soglia  $S$  aumentata.

### 3 Algoritmo di riconoscimento e primi test

L’algoritmo implementato può essere suddiviso in tre passi.

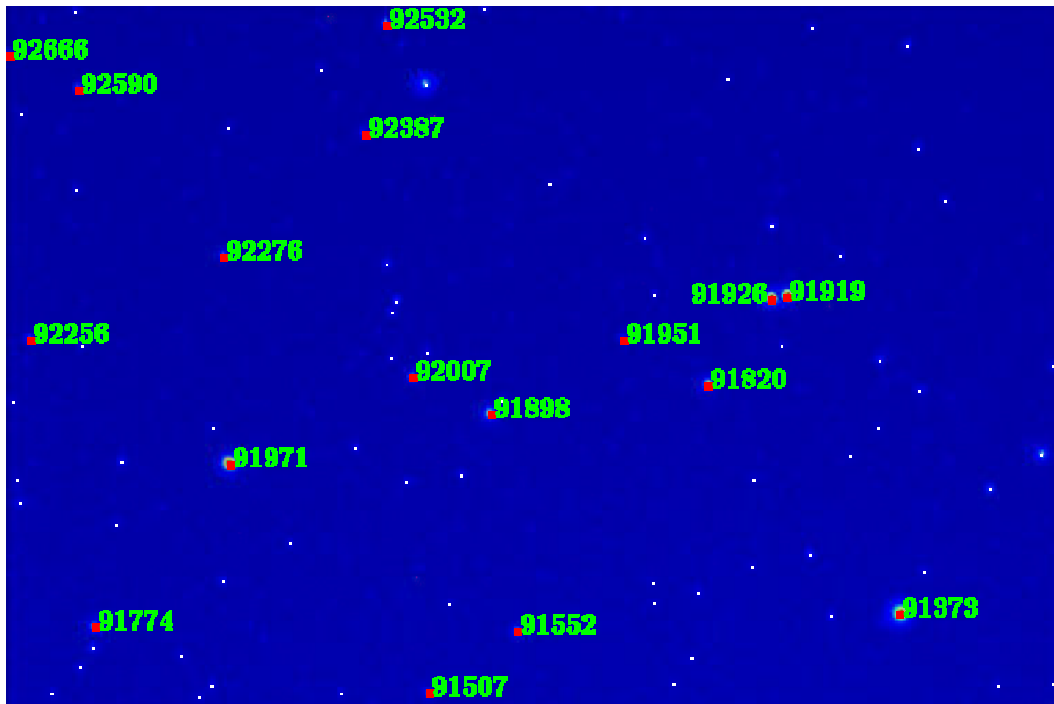
Per prima cosa, per ogni stella dell’immagine, si confrontano le distanze angolari delle stelle vicine. Presa una stella dell’immagine si calcolano le distanze angolari con le altre stelle dell’immagine. Successivamente, si confrontano queste distanze con le rispettive di ogni stella del catalogo, selezionando in caso di accordo di almeno due distanze, le possibili candidate.

Ridotte in questo modo le combinazioni, si passa al controllo sulle candidate: si verifica se concordano anche le distanze angolari relative tra le vicine della stella considerata, riducendo in questo modo il numero di possibilità.

Infine, partendo dalle candidate rimanenti, si costruiscono tutte le possibili combinazioni. Presa una candidata, si considerano le vicine “riconosciute” e si verifica se corrispondono ad altre candidate, procedendo così ad albero. Viene quindi scelta la combinazione con il numero maggiore di accordi.

Una volta riconosciute le stelle si calcola la direzione di puntamento del centro dell’immagine e il versore del lato maggiore dell’immagine.

Questo algoritmo è stato testato tramite un’immagine presa di un campo stellare noto. Si è utilizzato come sensore una CCD KAF400 con un’ottica di focale  $105\text{ mm}$  e un tempo di esposizione di  $10\text{ s}$ . L’immagine utilizzata con le stelle riconosciute, in accordo con quanto atteso, è mostrata in fig. 2.



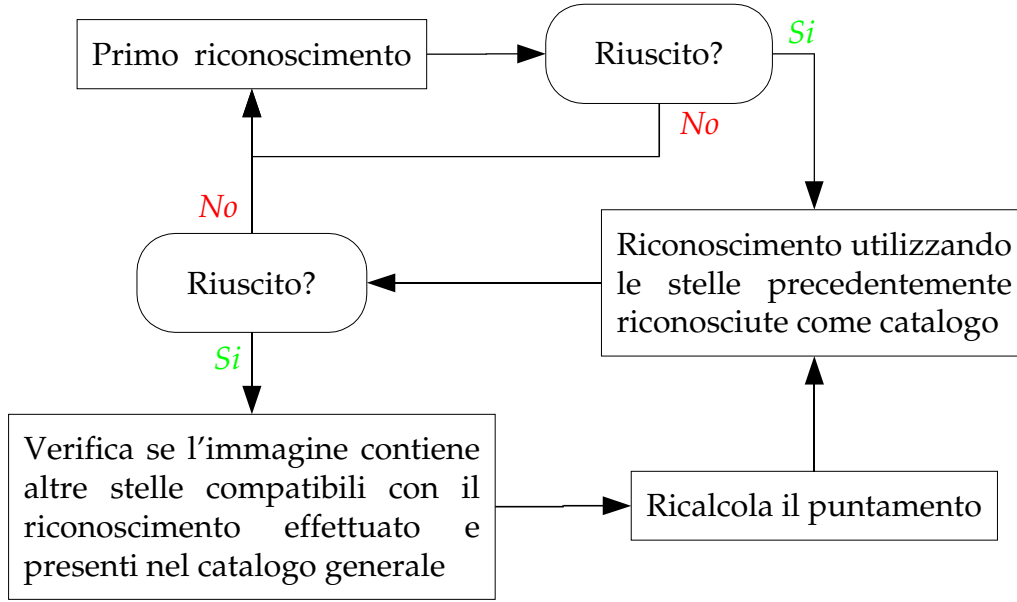
**Figura 2:** Immagine utilizzata per il test dell’algoritmo di riconoscimento: sono indicate le stelle riconosciute con il corrispettivo numero di catalogo hiparcos. L’immagine è stata acquisita con una focale di 105 mm e con 10 s di tempo di esposizione. Il catalogo utilizzato contiene stelle fino a magnitudo 8.

## 4 Algoritmo di inseguimento

Supponiamo di avere la telecamera di ripresa delle immagine in movimento. Una volta riconosciuto un campo stellare, si può cercare di “inseguire” il puntamento, basandosi sul riconoscimento precedente. Per far ciò si è implementato l’algoritmo in fig. 3.

Una volta che si riesce a riconoscere la direzione di puntamento dell’immagine, si costruisce un catalogo di stelle contenente le stelle riconosciute. Si utilizza quindi questo catalogo per il riconoscimento dell’immagine successiva. Se il tentativo ha successo si calcola la direzione di puntamento e, basandosi sul catalogo esterno, si verifica se si possono riconoscere altre stelle, migliorando così la precisione di puntamento e il numero di stelle riconosciute. Se, invece, il tentativo fallisce si passa al riconoscimento da zero.

Per verificare l’efficacia di questo algoritmo si sono utilizzate delle immagini simulate. Partendo da una direzione di puntamento, è stata costruita un’immagine supponendo che le stelle siano delle gaussiane. Tra un’immagine e l’altra è



**Figura 3:** Schema dell'algoritmo di inseguimento.

stato simulato un salto angolare pari a

$$\Delta\Theta = 10^{-2} \text{ rad} (\approx 15') \quad (1)$$

per l'ascensione e

$$\Delta\Phi = 5 \cdot 10^{-3} \text{ rad} (\approx 3') \quad (2)$$

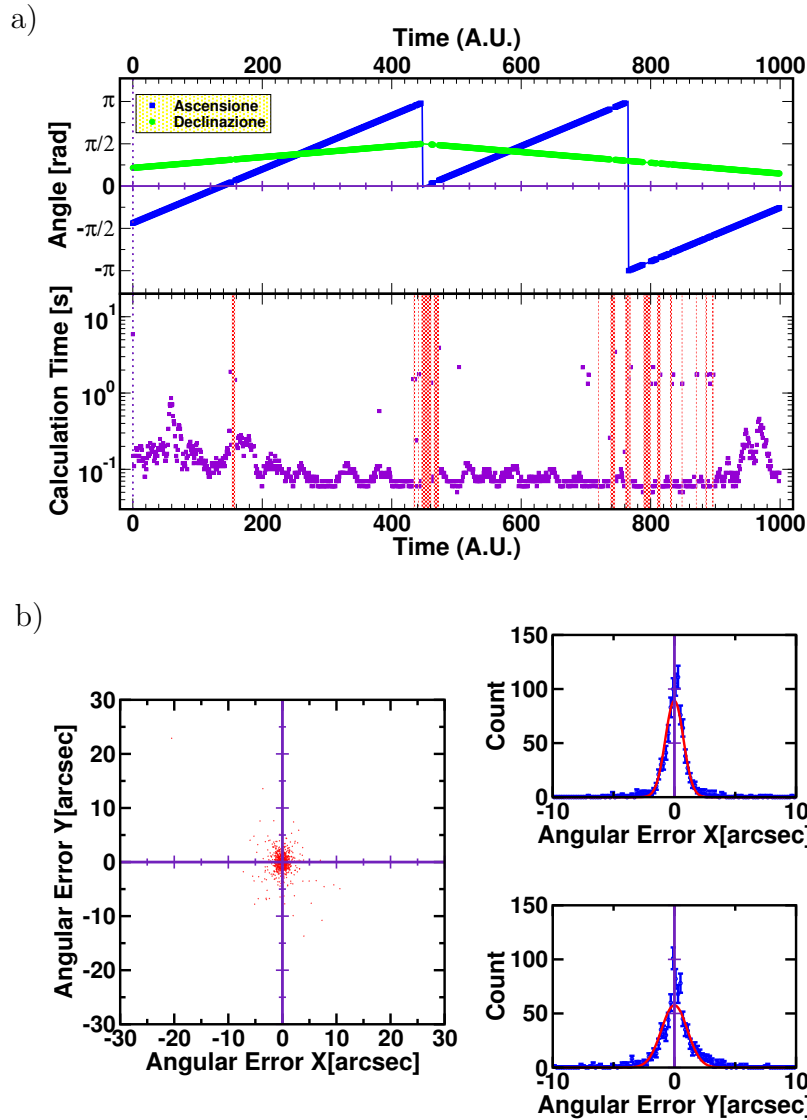
per la declinazione. Il puntamento simulato e ricostruito e il tempo di calcolo di ogni passo è riportato in fig. 4. In questa simulazione si è anche supposto nota la direzione di puntamento entro i  $0.1 \text{ rad}$ .

Come si può vedere dal grafico, il riconoscimento ha dato esito positivo nella maggioranza dei casi (929 su 1000). Dall'analisi dei tempi di calcolo dei passi si nota anche che l'inseguimento riduce notevolmente il tempo necessario per il riconoscimento, passando da qualche secondo a qualche decimo. Si nota inoltre che solo raramente l'algoritmo di inseguimento fallisce.

Analizzando le differenze tra il puntamento trovato e quello impostato si può avere indicazioni sulla precisione ottenibile. In fig. 4 vengono riportati gli scarti di questa simulazione e i relativi istogrammi.

Malgrado ci siano dei punti dispersi in un range di qualche decina di arcosecondo, la maggior parte è addensata nei pressi dello zero. Infatti, analizzando gli istogrammi degli scarti per i due assi dell'immagini e facendone i relativi fit con delle gaussiane si può trovare la precisione di puntamento ottenuta:

$$\sigma_X = 0.74 \text{ arcsec} \quad \sigma_Y = 1.07 \text{ arcsec} \quad (3)$$

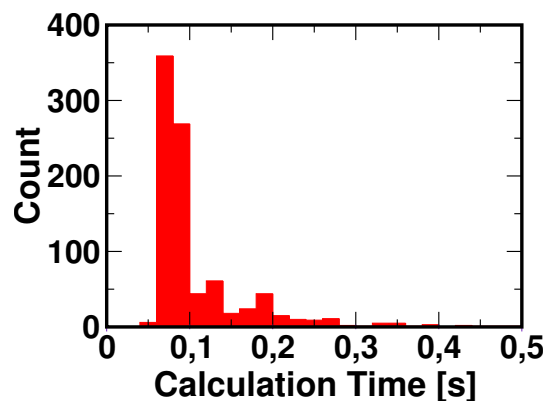


**Figura 4:** Risultato della simulazione di inseguimento: a - Coordinate di puntamento e tempo di calcolo; b - precisione di puntamento.

Infine, facendo un'istogramma dei tempi di calcolo dei passi, mostrato in fig. 5, si vede che la maggior parte delle volte è stato necessario un tempo inferiore a 0.1 s, da confrontarsi con il caso di riconoscimento da zero pari a 5.9 s.

## 5 Conclusioni

Allo stato attuale il programma, sia nella fase di riconoscimento che di inseguimento ha dato buoni risultati, richiedendo tempi di calcolo ragionevoli e



**Figura 5:** Istogramma dei tempi di calcolo della simulazione di inseguimento.

risultando abbastanza robusto.

Prossimamente si passerà ad eseguire dei test per l'inseguimento con immagini reali e si proverà il programma su un computer utilizzabile su palloni stratosferici.

## Riferimenti bibliografici

- [1] Curtis Padgett and Kenneth Kretz-Delgado, *Evaluation of Star identification techniques*, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California 91109
- [2] Laporte P., Abrosimov N., Bastie P., Cordier B., Cocco G.D., Evrard J., Gizzi L., Hamelin B., Jean P., Laurent P., Paltani P., Skinner G.K., Smither R.K. and von Ballmoos P., *CLAIRE - towards the first light for a gamma-ray lens*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, **442**, 438-442 (2000)
- [3] <http://astro.estec.esa.nl/SA-general/Projects/Hipparcos/hipparcos.html>

## Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
<b>2</b>	<b>Localizzazione delle stelle</b>	<b>2</b>
<b>3</b>	<b>Algoritmo di riconoscimento e primi test</b>	<b>3</b>
<b>4</b>	<b>Algoritmo di inseguimento</b>	<b>4</b>
<b>5</b>	<b>Conclusioni</b>	<b>6</b>