

# Identificação de Estrelas para Sensores Embarcados em Nanossatélites

Pedro Kukulka de Albuquerque<sup>1</sup>, Luis Eduardo Vergueiro Loures da Costa<sup>1</sup> e Valdemir Carrara<sup>2</sup>

<sup>1</sup>Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos/SP – Brasil

<sup>2</sup>Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos/SP – Brasil

**Resumo** – Este artigo visa desenvolver uma solução para a identificação estelar sem informação de atitude prévia no sensor de estrelas, de acordo com as restrições de energia e espaço interno presente em nanossatélites. Para isso foi utilizado dois algoritmos de identificação conhecidos como pirâmide e valor P, além de uma terceira opção formada por esses dois. Modificações na lógica clássica desses foram necessárias para atender as necessidades impostas ao problema. Por fim, o algoritmo foi simulado no Matlab em um ambiente ruidoso de forma a comparar as soluções quanto ao tempo para identificação e robustez ao cenário criado.

**Palavras-Chave** – Nanossatélites, Sensor de Estrelas, Identificação de Estrelas.

## I. INTRODUÇÃO

A busca por facilitar a exploração do espaço ocasionou uma tendência de diminuir o porte dos satélites a partir do início do século XXI. Essa mudança foi pautada na redução de custos e no tempo de desenvolvimento desses engenhos espaciais, como citado em [6]. Contudo, esse encolhimento gera desafios para adaptar a tecnologia dos componentes espaciais tradicionais para essa nova realidade, na qual há uma limitação de energia e espaço para acomodar esses equipamentos.

Esse problema torna-se latente no subsistema de determinação e controle de atitude, principalmente no contexto de missões que possuem requisitos de alta precisão, como em atividades de sensoriamento remoto óptico. Isso ocorre pois é imprescindível a utilização de rodas de reação e do sensor de estrelas para atingir demandas dessa natureza. Vale ressaltar que ambos equipamentos possuem um alto consumo energético e dimensões com alto impacto no espaço interno do satélite.

Dessa forma, este trabalho trata da problemática de miniaturização do sensor de estrelas, mais especificamente na parte de software. Assim, o objetivo é construir uma solução para identificação de estrelas que não necessite do conhecimento prévio da atitude da plataforma espacial, respeitando os requisitos advindos dos nanossatélites.

Para atingir esse objetivo, faz-se necessário explorar a composição de um sensor de estrelas, bem como o seu funcionamento.

### A. Sensor de Estrelas

O *hardware* desse sistema é basicamente composto por um defletor, elemento óptico, estrutura, sensor óptico e computador embarcado (Fig. 1). A função do defletor é

diminuir o ruído causado por fontes luminosas que atrapalham a determinação de atitude, como o Sol. Já o elemento óptico é composto por um conjunto de lentes cujo objetivo é projetar e corrigir a imagem no campo de visada do equipamento de forma adequada. A estrutura é responsável pelo controle térmico e pela proteção contra radiação. O sensor recebe os fótons e transforma esses dados em um sinal. No computador de bordo há o processamento do sinal recebido, transformando-o em informações de atitude.

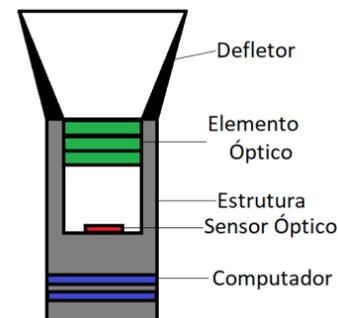


Fig. 1. Diagrama de um sensor de estrelas típico.

Baseado nos componentes acima descritos, o princípio de funcionamento do sensor de estrelas é capturar uma imagem de uma determinada seção do espaço, determinando a posição dos pontos luminosos capturados. Após, através de um processo de busca em um banco de dados, tenta-se identificar as estrelas na imagem. Dessa forma, na Fig. 2, pode-se observar esse processo resumido, o qual possui como finalidade obter a relação entre os eixos da plataforma e um sistema de referência inercial.

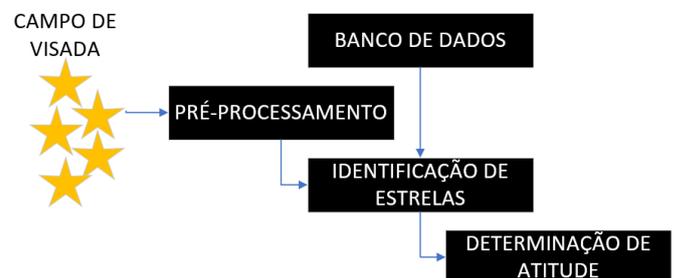


Fig. 2. Etapas para determinação de atitude utilizando um sensor de estrelas.

Como já comentado anteriormente, o foco desse artigo está na fase de identificação estelar. Essa fase ocorre de duas formas diferentes durante o funcionamento do sensor de estrelas. A primeira delas é quando se conhece as informações de atitude anterior do satélite e as utiliza para calcular o estado atual. Esse modo é chamado de acompanhamento, sendo melhor descrito em [2]. Já a segunda, que é o objetivo desse trabalho, ocorre quando não

se conhece nenhuma atitude inicial. Vale ressaltar, que esse último processo é mais lento que o primeiro e é utilizado quando se inicia o sensor de estrelas ou quando uma anomalia conduz esse equipamento a reiniciar o seu funcionamento.

## II. REVISÃO BIBLIOGRÁFICA

A etapa de identificação almeja atribuir uma estrela capturada no sensor a uma correspondente no banco de dados, através de semelhança entre características construídas baseadas no brilho e na posição relativa entre as estrelas na vizinhança. Existem diversos algoritmos na literatura capazes de realizar tais tarefas que, basicamente, se dividem, conforme [5], em dois grupos: reconhecimento de padrões e isomorfismo de subgrafos.

Os algoritmos de reconhecimento de padrões atribuem a cada estrela uma identidade única, isto é, um padrão. Assim, através de uma busca lógica, tenta-se identificar no banco de dados esse padrão que mais se aproxima da imagem capturada. Um exemplo dessa temática, desenvolvido em [4], é o algoritmo grade, que utiliza um quadriculado direcionado para criar uma palavra de bits para cada estrela, conforme Fig. 3. Nota-se que para cada quadrado com pelo menos uma estrela atribui-se o número 1, caso contrário utiliza-se o número 0. O padrão obtido por linhas da esquerda para direita é  $pat_m = 00000100...100$ . Assim, é possível percorrer o banco de dados utilizando uma busca lógica:

$$\max \left[ \sum_{i=1}^n \sum_{j=1}^n \{pat_m(i,j)\} \wedge \{pat_k(i,j)\} \right] \quad (1)$$

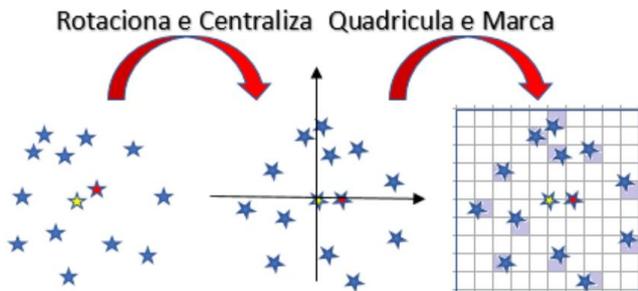


Fig. 3. Método para criar a palavra de identificação no algoritmo grade.

Segundo [4], esse processo, ao ser aplicado em equipamentos com grande abertura de campo de visada, resulta em uma alta taxa de sucesso na identificação (98%), mesmo em ambientes com fortes ruídos.

No caso dos algoritmos baseados no isomorfismo de subgrafos, a ideia é calcular a distância angular entre dois pontos luminosos capturados utilizando

$$D_{1,2} = \cos^{-1}(S_1 \cdot S_2) \quad (2)$$

e depois realizar a busca no banco de dados, sabendo que devido a ruídos é importante determinar uma tolerância  $\epsilon$ . Logo, o critério para identificar os candidatos é:

$$|D_{i,j} - D_{1,2}| < \epsilon \quad (3)$$

Esse tipo de algoritmo é muito suscetível a ruídos, além que para alguns valores de distância angular há diversos candidatos, como observado em [8]. Assim, diversas modificações surgiram para minimizar esse problema, o mais famoso é o descrito em [3], chamado de algoritmo pirâmide. Nesse, o objetivo é utilizar três valores de distâncias angulares para identificar três estrelas. Após, adiciona-se uma quarta estrela para confirmar a base previamente identificada (Fig. 4). O problema desse tipo de lógica é que em ambientes com muitas estrelas pode haver uma demora em identificar todas elas, já que há muitas possibilidades diferentes de se formar uma base triangular. Nesses casos a solução é truncar o número máximo de estrelas por imagem, como mostrado em [2].

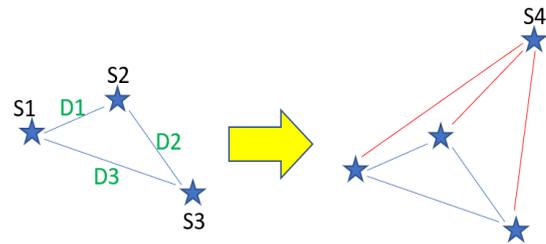


Fig. 4. Princípio de funcionamento do algoritmo pirâmide.

Outro algoritmo possível, do mesmo grupo que o anterior, é o do valor P, descrito em [7]. Nesse a finalidade é projetar um vetor do conjunto  $\mathbb{R}^3$ , com três distâncias angulares, no conjunto  $\mathbb{R}$ , com apenas um componente. Isso é realizado através de um produto escalar com o eixo de projeção principal ( $\Omega$ ) previamente determinado com todas as estrelas do banco de dados, ou seja, calcula-se o eixo no qual há um maior espalhamento dos pontos a serem projetados. Assim, para cada estrela capturada pelo sensor, encontra-se as duas vizinhas mais próximas dentro de uma faixa de distâncias (Fig. 5), formando um vetor  $\alpha$  com as distâncias angulares entre elas. Calcula-se o valor P medido através de

$$P_m = \Omega \cdot \alpha \quad (4)$$

e realiza-se uma busca desse valor no banco de dados considerando uma pequena tolerância  $\epsilon$  devido aos ruídos existentes de acordo com (5).

$$|P_m - P_k| < \epsilon \quad (5)$$

Esse algoritmo reduz o tempo de busca em relação ao da pirâmide, já que só é necessário buscar um único valor e não há a necessidade de calcular vários triângulos por cada estrela, conforme [8].

## III. RESTRIÇÕES EM NANOSSATÉLITES

Antes de construir uma solução para o objetivo desse trabalho baseado nos algoritmos apresentados, faz-se necessário enumerar quais os impactos das características de

uma missão que utiliza nanossatélites nos sensores de estrelas.

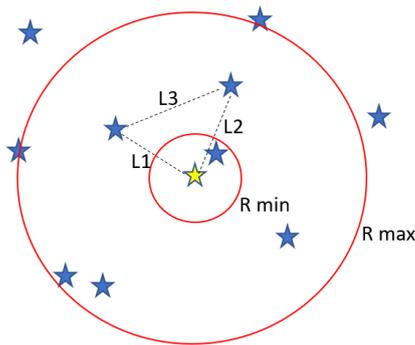


Fig. 5. Formação de triângulos no algoritmo valor P.

Conforme explicado em [2], nota-se que:

- O sensor deve ser autônomo, isto é, como o espaço é limitado ele não deve depender de outros para determinar a atitude do satélite. Logo, o ideal é que só haja um sensor de estrelas por satélite;
- O sensor deve exigir um baixo processamento, devido à energia limitada disponível. Isso impacta em um algoritmo simples que deve refletir em um rápido tempo de processamento;
- O sensor deve possuir um tamanho pequeno. Isso pode ser obtido diminuindo o campo de visada dos sensores, implicando em menos estrelas capturadas por imagem;
- O sensor deve possuir componentes com um custo reduzido. Isso ocasiona em sensores ópticos não muito precisos quanto ao brilho da estrela; e
- O sensor deve ser robusto para suportar ruídos, devido ao fato de não depender de outros sensores e de ser o elemento de determinação de atitude principal. Logo, a parte lógica do sensor deve ser capaz de suportar esse tipo de problema autonomamente.

De acordo com os pontos levantados e o objetivo desse artigo, conclui-se que o algoritmo desenvolvido deve:

- Basear-se em uma única imagem por medida no tempo;
- Possuir um código que exija um baixo processamento;
- Ser capaz de trabalhar com poucas estrelas;
- Não utilizar o brilho estelar como fonte de identificação; e
- Possuir uma lógica que suporte os ruídos do ambiente.

#### IV. ALGORITMOS E CENÁRIOS

Nessa seção o objetivo é verificar quais algoritmos serão adaptados e testados para o caso dos nanossatélites. Como visto, os algoritmos de identificação dividem-se em dois grupos principais. Com relação aos da classe de

reconhecimento de padrões foi visto que eles foram criados para atender um cenário com uma grande abertura de campo de visada, isto é, mais estrelas captadas por imagem. Nota-se que esse não é o caso da plataforma espacial alvo desse artigo e, como comentado em [2], os resultados não são adequados nesse caso. Isso ocorre pois quando há pouca quantidade de estrelas por imagem o algoritmo não chega a uma conclusão válida devido a ambiguidade causada por poucos dados de comparação, além de ser muito sensível a ruídos. Nesse sentido, o algoritmo para esse artigo deve ser construído baseado nas distâncias angulares entre as estrelas.

Dessa forma, respeitando as restrições descritas na seção anterior, foram determinados os algoritmos pirâmide e valor P como fonte confiáveis para esse projeto. O pirâmide foi escolhido devido a ser robusto a ruídos e o valor P devido ao rápido tempo de identificação. Além disso, há uma terceira opção que é uma mistura dos dois anteriores de forma a aproveitar as vantagens de ambos, sendo chamado nesse trabalho de algoritmo misto.

A forma clássica do algoritmo pirâmide previsto em [3] deve ser modificada para atender as peculiaridades dos nanossatélites (Fig. 6). Para diminuir o tempo de processamento foi definido um número máximo de estrelas investigadas por imagem, no caso as mais brilhantes que geram posições mais confiáveis. Além disso, para dar características de robustez, utiliza-se o algoritmo TRIAD, desenvolvido em [1], para confirmar a base do triângulo além da adição da quarta estrela. Assim, tenta-se identificar a quarta estrela com duas estrelas da base de forma rápida e simples, aumentando a confiabilidade do processo. Para isso, basta utilizar os dois vetores de posição das estrelas observados  $\{u_1, u_2\}$  com os dois no referencial inercial determinados no banco de dados após a identificação prévia  $\{v_1, v_2\}$  para determinar a matriz de transformação  $A$ , conforme (6) a (8).

$$[r_1, r_2, r_3] = [v_1, v_1 \times v_2, v_1 \times (v_1 \times v_2)] \quad (6)$$

$$[s_1, s_2, s_3] = [u_1, u_1 \times u_2, u_1 \times (u_1 \times u_2)] \quad (7)$$

$$A = \sum_{i=1}^3 s_i r_i^T \quad (8)$$

A quarta estrela pode ser finalmente confirmada utilizando

$$A v_4 = u_4 \quad (9)$$

onde  $v_4$  é o vetor posição no banco de dados após identificação prévia e  $u_4$  é o vetor observado. Por fim, verifica-se se há alguma estrela que possui uma distância maior do que o campo de visada consegue capturar dentre todas as identificadas na imagem.

Para o caso do algoritmo valor P implementado em [7] também há as mesmas modificações descritas para o pirâmide, de forma a atender todos os requisitos da seção anterior (Fig. 7). A única diferença é que o algoritmo TRIAD



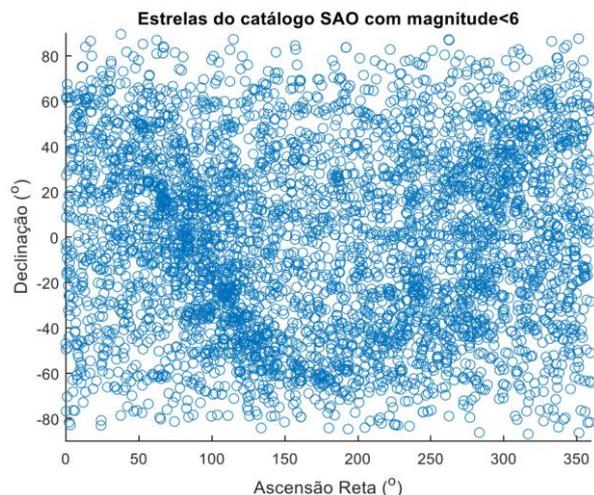


Fig. 9. Catálogo de estrelas SAO J2000 com magnitude menor que 6.

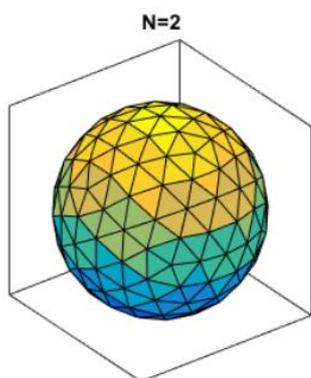


Fig. 10. Icosaedro subdividido duas vezes.

Assim, para criar um ambiente de simulação mais parecido com o real, foi necessário adicionar ruídos diretamente nas estrelas contidas na Fig. 9 que variavam entre os seguintes valores:

- Posição da estrela (x,y) na imagem capturada pelo sensor: 0 a 2 pixels;
- Magnitude da estrela: 0 a 1 de variação na magnitude visual;
- Adição de estrelas na imagem (simulando radiação e planetas): até duas estrelas virtuais por imagem;
- Remoção de estrelas na imagem (ocorre quando há muito ruído no sensor): até duas estrelas por imagem.

Usando um gerador de números aleatórios no Matlab, foram criadas 1000 imagens de diferentes regiões da esfera celeste com os ruídos já descritos adicionados.

As características do sensor óptico utilizado na simulação estão presentes na Tabela I.

Vale ressaltar que todos os códigos foram implementados no *software* Matlab e rodados em um computador com configuração com processador *core* I7 e 8 Gb de memória RAM.

Tabela I. Características do sensor óptico.

<b>Pixel Resolution</b>	1024 x 1024
<b>Pixel Size</b>	12 $\mu$ m x 12 $\mu$ m
<b>Principal Point (x,y)</b>	(0,0)
<b>Focal Length</b>	58,4536 mm
<b>FOV</b>	12 $^{\circ}$ x 12 $^{\circ}$

## V. RESULTADOS E DISCUSSÕES

Os resultados podem ser observados na Fig. 11 a Fig. 14. Para iniciar a discussão deve-se ressaltar que um bom algoritmo nesse caso é um que apresente alta taxa de identificação com sucesso, com menor erro possível e com um tempo de execução aceitável.

Assim, como era esperado o tempo do algoritmo misto é maior, porém todos os três possuem a mesma ordem de grandeza. Portanto, esse critério não é decisivo.

Já a taxa de identificação com sucesso é maior no algoritmo pirâmide e ligeiramente menor no misto. Essa diminuição ocorreu, pois, a lógica do valor P criou dúvidas no processo, para ser mais exato isso só ocorreu em uma única imagem. As duas estrelas adicionadas foram em um local no qual deveria haver uma estrela, confundindo o programa.

Por fim, nota-se que só ocorreu erro em uma única imagem para o algoritmo misto e em duas no pirâmide. Na verdade, não foi propriamente um erro. Todos esses casos são referentes a situações no qual foi adicionado uma estrela nas proximidades do local que foi retirada outra e a determinação de atitude poderia ocorrer normalmente, pois o triângulo real foi corretamente identificado. No caso do algoritmo valor P, ocorreram erros de identificação.

Nesse sentido, o algoritmo pirâmide possui uma ligeira vantagem temporal, bem como na taxa de identificação com sucesso. Já o misto possui uma ligeira vantagem com relação aos erros. Logo, pode-se dizer que ambos cumprem o objetivo de forma adequada.

Entretanto há um outro fator não citado que pode diferenciar os resultados e está relacionado com o número de estrelas identificadas por imagem. Quanto maior número de corpos celestes determinados, maior será a precisão da atitude, fato citado em [8]. Assim, os resultados referentes a esse aspecto são apresentados na Fig. 14, mostrando a vantagem de uma solução híbrida.

## VI. OBSERVAÇÕES FINAIS

A necessidade de miniaturização do sensor de estrelas para cumprir os requisitos dos nanossatélites gera desafios a serem vencidos e, neste artigo, foram trabalhados somente no aspecto da lógica para a identificação de estrelas no modo de operação sem informação de atitude prévia.

Nesse sentido, buscou-se explorar os algoritmos baseados no isomorfismo de subgrafos, isto é, os que utilizam a distância angular entre as estrelas. Isso é necessário pois os baseados em reconhecimento de padrões necessitam de

muitas estrelas por imagem para funcionar com todas as suas potencialidades, o que não ocorre nos nanossatélites.

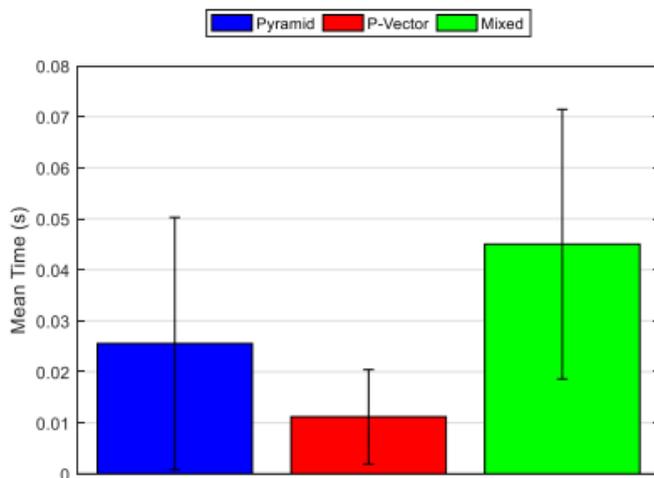


Fig. 11. Tempo médio de execução dos algoritmos com desvio padrão.

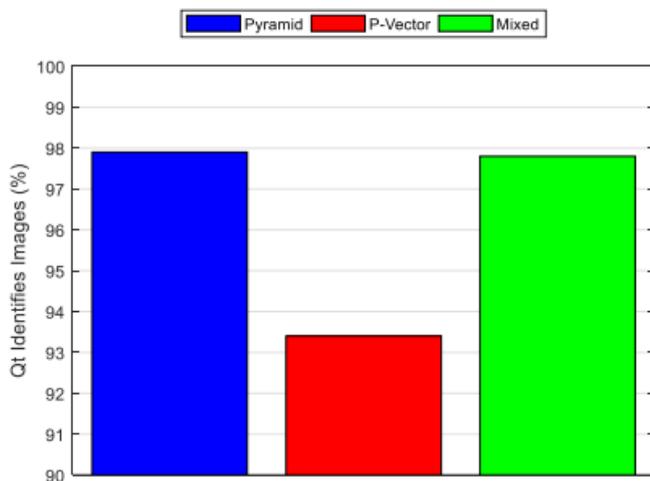


Fig. 12. Porcentagem de imagens identificadas, isto é, cada imagem com no mínimo 4 estrelas corretamente identificadas e sem erros de identificação.

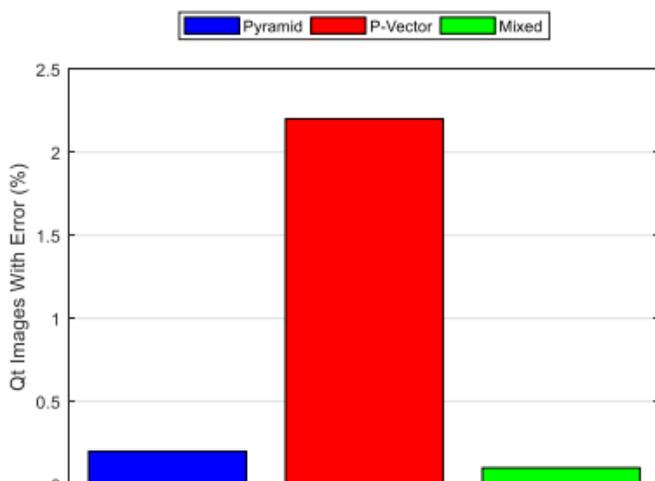


Fig. 13. Porcentagem de imagens que contenham ao menos 1 estrela identificada erroneamente.

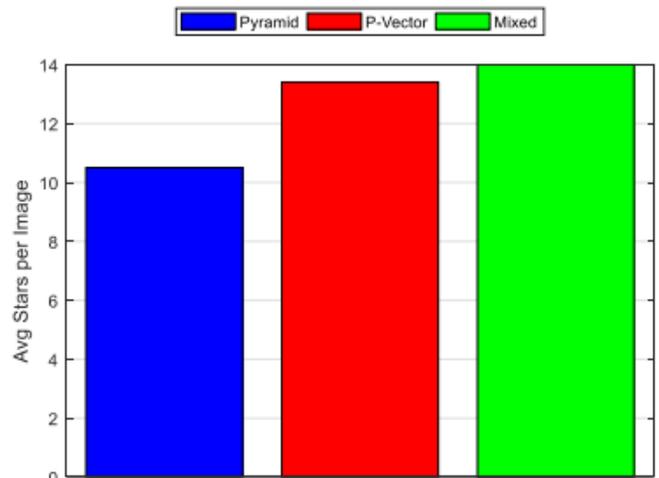


Fig. 14. Quantidade de estrelas identificadas por imagem.

Os dois algoritmos escolhidos de acordo com as restrições derivadas da limitação de energia e de espaço interno foram o valor P e o pirâmide, ambos com modificações adequadas a situação.

Além disso, utilizou-se uma terceira opção utilizando os pontos fortes desses dois representantes do isomorfismo, chamado de algoritmo misto.

Os resultados mostraram que o valor P possui um desempenho inferior ao se considerar o tempo médio de identificação e a robustez aos ruídos. Já os outros dois são equivalentes.

O que pode determinar a utilização de um em detrimento do outro é se a precisão esperada na atitude for muito alta, dessa forma a melhor opção seria o algoritmo misto. Isso ocorre pois é possível obter mais identificações por imagem.

Por fim, a sugestão é que sejam estudados a influência das limitações dos nanossatélites nas outras etapas do software do sensor de estrelas de forma a adaptar a solução aqui proposta.

## REFERÊNCIAS

- [1] H. D. Black, "A passive system for determining the attitude of a Satellite", AIAA Journal, vol. 2, n. 7, p. 1350-1351, 1964.
- [2] P. K. De Albuquerque, "Algoritmo de identificação e determinação de atitude para sensor de estrelas aplicável a pequenos satélites", 2017, 187 f. Trabalho de Conclusão de Curso (Engenharia Aeroespacial) – Instituto Tecnológico de Aeronáutica, São José dos Campos, 2017.
- [3] D. Mortari, M. A. Samaan, C. Bruccoleri, J. L. Junkins, "The pyramid star identification technique", Journal of the Institute of Navigation, vol. 51, n. 3, p. 171-183, 2004.
- [4] C. Padgett, K. Kreutz-Delgado, "A Grid algorithm for autonomous star identification", IEEE Transactions on Aerospace and Electronic Systems, vol. 33, n. 1, p. 202-213, Janeiro 1997.
- [5] E. Silani, M. Lovera, "Star identification algorithms: novel approach and comparison study, IEEE Transactions on Aerospace and Electronic Systems, vol. 42, n. 4, p. 1275-1288, Outubro 2006.
- [6] J. R. Wertz. Introduction. In: \_\_\_\_\_. "Space mission engineering: the new SMAD". Hawthorne: Microcosm Press, 2011. cap. 1, p. 1-28.
- [7] J. Yang, G. Zhang, J. Jiang, "Fast star identification algorithm using P vector", Acta Aeronautica, vol. 28, n. 4, p. 897-900, 2007.
- [8] G. Zhang, "Star identification: methods, techniques and algorithms", Berlin: Springer-Verlag, 2017, 223 p.