Using Shape Descriptors for Star Identification

Cedric Liman, **Sergio Montenegro**, **Nikolaj Bernhardt**
Chair of Computer Science 8: Aerospace Information Technology
University of Würzburg, Germany
cedric.liman@uni-wuerzburg.de

Abstract

Spacecraft attitude determination plays an important part in most missions. Star trackers are one of the most reliable and accurate sensors used for this task. One of the main problems in a star tracker is the identification of the stars in the image to determine the spacecraft's attitude. So far, a multitude of algorithms have been developed to identify stars and the patterns they form. In this work, we propose to use several well-known pattern recognition techniques in a star tracker to identify the stars. Specifically, Hu moments, complex moments, Zernike moments, and Fourier descriptors are applied to the star identification problem for the first time. These are proven methods used to identify various patterns in images, like printed characters, people, or sign language. To evaluate their performance, they were tested inside a full star tracker algorithm which includes star extraction, centroiding, star identification, verification, and attitude computation. As a reference, the triangle method from Liebe [1] is used. Compared to this, only Fourier descriptors offer similar performance in terms of accuracy, robustness, runtime, and memory requirements. Hu moments and complex moments perform slightly worse than the reference method in all metrics, except for high levels of noise in the image. Zernike moments perform the worst out of all the tested methods. All tests were conducted on both simulated images and real images.

1 Introduction

The attitude of a spacecraft describes how it is oriented in 3D space relative to a reference coordinate system. Knowing and controlling the attitude of a spacecraft is an important task in most missions, since it has an effect on nearly all subsystems, like power, thermal, or communication. Therefore, measuring the attitude of a spacecraft is an essential part of almost every mission. Star trackers generally provide the most accurate method of measuring the spacecraft's attitude, and they offer an absolute measurement with three degrees of freedom [2]. This makes them a popular choice on a spacecraft. In order to determine the orientation, they need to be able to identify the stars in the camera's limited field of view (FOV) over the entire celestial sphere. For this task, various algorithms have been developed over the years and became increasingly efficient and robust [3, 4]. The main problem can be broken down to the recognition of a known pattern in an image. This is the main subject in the field of pattern recognition, and has been solved for numerous kinds of tasks [5–7]. Both fields - star trackers and pattern recognition - have developed separate techniques and algorithms for solving the same problem of identifying a pattern in an image. In this work, known techniques from the field of pattern recognition will, for the first time, be applied to star trackers. Hu moments, complex moments, Zernike moments and Fourier descriptors are going to be used to identify star patterns. The newly proposed methods will be compared to the common and well established triangle method from Liebe [1]. An algorithm for a complete star tracker will be developed, where pattern recognition techniques can be compared to traditional star identification techniques.

2 Related Work

One of the earliest modern star identification methods was presented by Liebe [1], and enabled star trackers to function over the entire sky without prior knowledge of the initial attitude. It uses a triangle pattern constructed out of a central star, its two closest neighbors, and the angles between them. A different approach at creating the pattern was presented by Padgett and Kreutz-Delgado [8]. Their pattern is constructed by placing a grid of uniformly space squares on top of the image. Each grid cell containing at least one star is marked as "on" and the others "off". The positions of all "on" cells then form the pattern. An improvement to the grid algorithm was presented by Na et al. [9]. It matches the generated pattern to the database using a different cost function, which allows a star to be in a different grid cell if

some noise is present in the position. Furthermore, a cell is not "on" or "off" but instead has the value of the magnitude of the star in this cell. These two improvements increase the accuracy of the grid based approach but come at the cost of increased processing time. The so called *Search-Less* algorithm from Mortari [10] primarily focuses on improving search time. Only the angular distance between two stars is used to identify them. With this pattern every possible star pair combination in the image is checked. The improvement in search time comes from sorting all angular distances in ascending order and fitting a linear function to them. While this linear function is not a perfect fit to the data, it can be evaluated quickly. Only a small area around the value of this function then needs to be searched. Mortari et al. [11] improved this idea with the *Pyramid* algorithm. It is a pattern of four stars where the angular distance between every star pair is used for identification. The angular distances are searched with the same method. This algorithm has the advantage, that it works reliably even if lots of false stars are present in the image. A different pattern was used by Silani and Lovera [12] in their *Polestar* algorithm. Here, all stars around a central star are considered, which have an angular distance from the central star between a fixed interval. The angular distances are then discretized into different bands around the central star. A binary sequence is formed based on whether there is a star present inside the band or not. This sequence identifies the pattern.

3 Shape descriptors for star identification

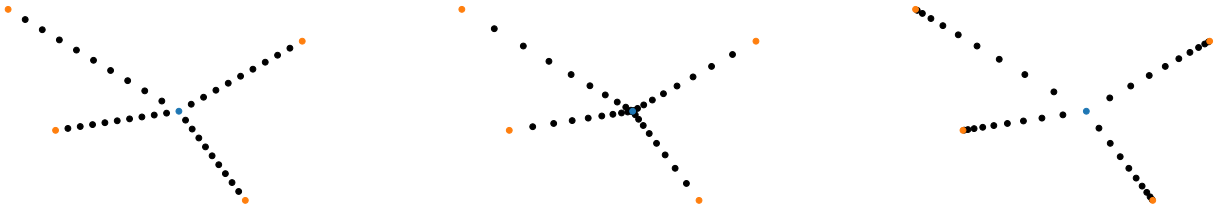
In the following section it is described, how different shape descriptors can be used for star identification. It is illustrated, how a single star pattern can be identified using these techniques. In a star tracker, this step is repeated for every star pattern in an image and the results can be verified against each other. This can lead to only a partial identification of the stars in an image or no identification at all. During the evaluation in section 4, this is considered, but not explained in more detail in this work.

3.1 Hu Moments

Hu moments consist out of seven parameters describing a 2D pattern and were introduced by Hu [13]. Their most prominent property is, that they describe a pattern regardless of its position, orientation, or scale in an image. They were first used by Hu to identify different, printed alphabetic characters in an image. Later Hu moments were applied to many tasks [5, 14, 15]. Despite their wide adoption, they have never been used in star trackers even though their properties make them well suited for this task. Depending on the spacecraft's attitude, the pattern the stars form can be in different positions in the image and rotated differently.

To compute the Hu moments, first, a pattern has to be created. Many unique patterns can be created to compute the Hu moments of, but all of them should adhere to some constraints. One main star needs to be chosen, which gets associated with the unique Hu moments of this pattern. At least one additional star, preferable more, have to be selected to form a unique pattern. These added stars should be close to the main star, since otherwise they could be outside the FOV in case the main star is close to the edges of the image. Though, there is a trade-off for the number of added stars. More added stars contain additional information, which helps in the identification. On the other hand, they require more storage and computational resources, and if one star is missing, the entire pattern may become unusable. The most obvious choice for them are the closest neighbors of the main star. Ergo, the pattern should consist out of a central star and a number of n neighbors around it. Furthermore, the pattern itself must be able to be constructed regardless of the position and rotation of the stars. The pattern proposed here contains the positions of the central star and its n neighbors. Additionally, lines are drawn from the central star to each of its neighbors. The line has to be sampled to be used as pattern. Three sampling types are explored here: linear sampling, quadratic sampling, and inverse quadratic sampling. A visual comparison between the three different sampling methods can be seen in Figure 1. The used patterns consist out of the coordinates of the central star, the coordinates of its n neighbors and the points of the sampled lines between them. While the brightness of a star could also be included in the pattern, it is often neglected, since a precise measurement of it is difficult to achieve with a star tracker. On the contrary, the measurement of the position of the stars is more reliable and accurate [10]. Computing the Hu moments starts by computing the raw moments defined by equation (1).

$$M_{pq} = \sum_{x} \sum_{y} x^p y^q I(x, y) \tag{1}$$



(a) Linear sampled lines.

(b) Quadratic sampled lines. An emphasis is put on the central star.

(c) Inverse quadratic sampled lines. An emphasis is put on the neighboring stars.

Figure 1: Three different, possible samplings of a line to emphasize different parts of the formed pattern.

The raw moment M_{pq} has the order $(p+q)$ and adds up the coordinates of each point (x, y) and its intensity $I(x, y)$. For the presented patterns, this is the sum over all points in the pattern. The intensity of each point is always $I(x, y) = 1$. The raw moments can be used to compute the center of the pattern $\left(\bar{x} = \frac{M_{10}}{M_{00}}, \bar{y} = \frac{M_{01}}{M_{00}}\right)$. After the raw moments, the central moments μ_{pq} need to be computed with equation (2).

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \tag{2}$$

By subtracting the center of the pattern, the central moments are invariant to translation. Hu proposed a specific combination of central moments to also achieve rotation-invariance. The seven Hu moments $I_1 ... I_7$ can be calculated using equation (3).

$$\begin{aligned}
I_1 &= \mu_{20} + \mu_{02} \\
I_2 &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \\
I_3 &= (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \\
I_4 &= (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2 \\
I_5 &= (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] \\
&\quad + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \\
I_6 &= (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}) \\
I_7 &= (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] \\
&\quad - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2]
\end{aligned} \tag{3}$$

Since the Hu moments can have a large variation in magnitude, they are transformed to a logarithmic scale $h_i = \text{sign}(I_i) \log_2(|I_i|)$

3.2 Complex Moments

Flusser [16] proposed using complex moments for pattern recognition, a generalization of rotation invariant moments to higher orders, similar to Hu moments from section 3.1. Complex moments on their own have yet to find widespread adoption, due to their similarity to Hu moments, and are often used alongside Hu moments for higher orders [7, 17]. A complex moment c_{pq} of order $(p+q)$ is defined with equation (4), similar to how the central moments are defined in equation (2). This makes them already translation invariant.

$$c_{pq} = \sum_x \sum_y [(x - \bar{x}) + i(y - \bar{y})]^p [(x - \bar{x}) - i(y - \bar{y})]^q I(x, y) \tag{4}$$

Here, i represents the imaginary unit. Flusser proves, that rotation invariance can be achieved using complex moments with the following rule: For $n \geq 1$ and for non-negative integers k_i, p_i and q_i with $i = 1, ...n$ which combine to 0 according to equation (5), rotation invariance can be achieved through the combination shown in equation (6).

$$\sum_{i=1}^{n} k_i (p_i - q_i) = 0 \tag{5}$$

$$\prod_{i=1}^{n} c_{p_i q_i}^{k_i} \tag{6}$$

An example of this rule would be c_{11}, $c_{20}c_{02}$ or $c_{20}c_{12}^2$. For 2D pattern recognition, Flusser suggests using the combinations of complex moments as seen in equation (7), up to the fourth order. Even higher order moments could be constructed, but there is a trade-off between higher order moments to capture finer

details, and lower order moments, which are less susceptible to noise.

$$\begin{array}{lll} C_1 = c_{11} = I_1 & C_5 = \mathrm{Re}\left[c_{30}c_{12}^3\right] = I_5 & C_9 = \mathrm{Im}\left[c_{31}c_{12}^2\right] \\ C_2 = c_{21}c_{12} = I_4 & C_6 = \mathrm{Im}\left[c_{30}c_{12}^3\right] = I_7 & C_{10} = \mathrm{Re}\left[c_{40}c_{12}^4\right] \\ C_3 = \mathrm{Re}\left[c_{20}c_{12}^2\right] = I_6 & C_7 = c_{22} & C_{11} = \mathrm{Im}\left[c_{40}c_{12}^4\right] \\ C_4 = \mathrm{Im}\left[c_{20}c_{12}^2\right] & C_8 = \mathrm{Re}\left[c_{31}c_{12}^2\right] & \end{array} \tag{7}$$

As visible in equation (7), there are some similarities between the Hu moments and the complex moments for pattern recognition. For star identification, the same steps as for Hu moments from section 3.1 are performed. The identical pattern is constructed, but instead of computing the Hu moments, the complex moments from equation (7) up to the desired order are calculated. The scaling is the same as for Hu moments.

3.3 Zernike Moments

Teague [18] was the first one to suggest using orthogonal moments on the basis of orthogonal polynomials. The orthogonal polynomials he chose were the Zernike polynomials, proposed by Zernike [19], since they were invariant to rotation. Therefore, the name Zernike moments. Zernike moments were also used successfully in different applications [14, 20, 21]. Zernike polynomials are a set of complex, orthogonal polynomials, defined on the unit disk. The polynomials are only rotation-invariant. To also achieve translation-invariance the pattern first needs to be centered on the image plane. Therefore, the central star is positioned in the center of the image plane and the neighbors are shifted accordingly. In the case of star trackers, the same patterns are used as for Hu moments and complex moments in sections 3.1 and 3.2. For the Zernike polynomials to be used as image moments, the image plane first needs to transformed into the unit disk [20]. To compute the Zernike moments, one starts off with the radial polynomial $R_{n,m}$, defined in equation (8).

$$R_{n,m}\left(r\right) = \sum_{s=0}^{(n-|m|)/2} \left(-1\right)^s \frac{\left(n-s\right)!}{s!\left(\frac{n+|m|}{2}-s\right)!\left(\frac{n-|m|}{2}-s\right)!} r^{n-2s} \tag{8}$$

n is a non-negative integer and represents the order of the radial polynomial, m is a positive or negative integer. m has to follow the constraints $n - |m| =$ even and $|m| \leq n$ and represents the repetition of the azimuthal angle. The radial polynomial of fixed order only depends on the radius r. To construct the Zernike moments, the Zernike polynomials $V_{n,m}(r,\theta) = R_{n,m}(r)e^{i\theta m}$ are defined in the unit disk with polar coordinates using the radial polynomials $R_{n,m}(r)$. The Zernike moments sum over the Zernike polynomial for every point in the pattern of size λ and normalize them, which can be seen in equation (9).

$$Z_{n,m} = \frac{n+1}{\lambda} \sum_x \sum_y I(x,y) V_{n,m}^*(r(x,y),\theta(x,y)) = \frac{n+1}{\lambda} \sum_x \sum_y I(x,y) R_{n,m}\left(r(x,y)\right) e^{-i\theta(x,y)m} \tag{9}$$

$V_{n,m}^*$ is the complex conjugate of the Zernike polynomial. For the identification of the pattern, the magnitude of the complex Zernike moment from equation (9) has to be taken to achieve rotation-invariance, and it gets converted to logarithmic scale using the logarithm of basis 2.

As a variation to the original Zernike polynomials, Bhatia and Wolf [22] proposed the so-called pseudo Zernike polynomials. They should be more resilient to noise, according to Otiniano-Rodrıguez et al. [14]. The difference lies in a modified radial polynomial $R_{n,m}^p$, which can be seen in equation (10).

$$R_{n,m}^p = \sum_{s=0}^{n-|m|} \left(-1\right)^s \frac{\left(2n+1-s\right)!}{s!\left(n-|m|-s\right)!\left(n+|m|-s+1\right)!} r^{n-s} \tag{10}$$

For the pseudo Zernike moments, n is still a non-negative integer, but m can be a positive or negative integer, only constraint by $|m| \leq n$.

3.4 Fourier Descriptors

Fourier descriptors are used to identify shapes with a closed contour and were first proposed by Cosgriff [23]. They also provide translation-, scale- and rotation-invariance like the other presented methods,

and thus they are a good choice for pattern recognition. Their main difference is that they work on the contour of the pattern instead of the pattern itself. However, they still found widespread adoption for shape identification [5, 24, 25]. For the application of Fourier descriptors to star trackers, a contour pattern needs to be created. Starting with a central star, its n closest neighbor are selected. Then, a line is drawn, starting from the central star, going to the closest neighbor. The line continues from the closest neighbor to the second-closest neighbor and so on. To close the contour, the last line segment goes from the most distant of the n neighbors back to the central star. For processing, the lines need to be sampled in regular intervals, which can be seen in Figure 2.

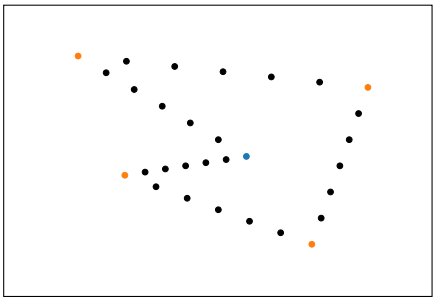


Figure 2: Sampled contour pattern used for Fourier descriptors. The blue dot is the central star and the orange ones the $n = 4$ neighboring stars.

To compute the Fourier descriptors, the discrete Fourier transformation is applied to the sampled points, which can be seen in equation (11).

$$f_k = \sum_{j=0}^{N-1} (x_j + iy_j) \exp\left(\frac{-2\pi ijk}{N}\right) \tag{11}$$

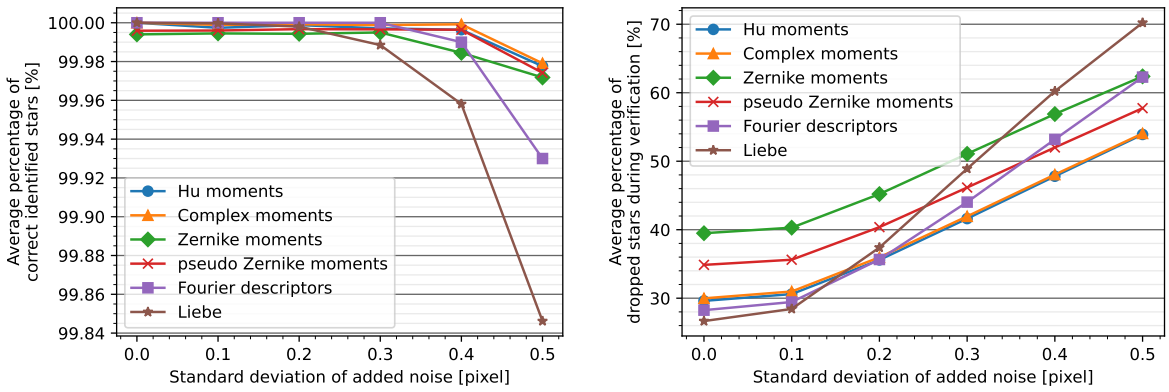
This results in N complex Fourier coefficients f_k, $k = 0, ..., N-1$ for N sampled points in the contour pattern. x_j and y_j are the coordinates of these points. f_0 only contains the position of the pattern, and can therefore be discarded. When taking the magnitude of the complex Fourier coefficients, they become invariant to rotation, since a rotation only effects the phase of the coefficient according to Conseil et al. [5]. Scale invariance can be achieved by normalizing the coefficients $F_k = \frac{|f_k|}{|f_1|}$, $k = 2, ..., N-1$ with f_1. While the invariance to scaling is not needed for star trackers, the normalization improves their resilience to noise, and is therefore still used here. This results in the normalized Fourier descriptors F_k, $k = 2, ..., N-1$ being used here. Even though there are $N-2$ Fourier descriptors available, often only a subset of them is used for pattern recognition. There is a trade-off between capturing more details using higher order Fourier descriptors and their resilience to noise, which increases for lower order descriptors.

4 Evaluation

To evaluate the star tracker's performance over the entire sky, a Monte-Carlo simulation is used, sampling 10000 random spacecraft attitudes. For each sampled position, a simulated star image is generated, including noise and one disturbance star. A disturbance star is a star which was deleted, moved or added randomly, to account for occlusions or falsely classified stars. The new star identification algorithms are tested first using the ideal positions of the stars in the image with added noise to it. Then, in a full star tracker algorithm, where the stars have to be extracted from the image first and after their identification the attitude is computed and finally the same test is performed on real images. The five presented star identification methods have various parameters which influence their performance. These parameters were optimized by hand through trial and error. This results in them possibly not representing the optimal solution, since the parameter space is large or even infinitely large for same techniques. The final parameters selected for the test can be seen in Table 1. Knowing the selected parameters allows one

Method	**Identifiers**	**Number of stars in the pattern**	**Line interpolation method**	**Number of samples per line**
Hu moments	$h_1 ... h_7$	3	quadratic	2
Complex moments	$C_1 ... C_6$	3	quadratic	2
Zernike moments	$Z_{3,1}$, $Z_{4,2}$, $Z_{8,0}$, $Z_{8,4}$, $Z_{9,3}$, $Z_{10,0}$	3	inverse quadratic	2
pseudo Zernike moments	$Z^p_{3,1}$, $Z^p_{4,1}$, $Z^p_{5,1}$, $Z^p_{6,1}$, $Z^p_{7,2}$, $Z^p_{8,2}$, $Z^p_{9,2}$, $Z^p_{10,2}$, $Z^p_{10,3}$	3	inverse quadratic	2
Fourier descriptors	$F_2 ... F_4$	3	linear	5

Table 1: Selected parameters for the star identification algorithms.



(a) Percentage of correctly identified stars under various levels of added noise.

(b) Percentage of identified stars in the image under various levels of added noise.

to directly estimated the memory requirements of each technique, since the many differentiating factor is the number of identifiers which need to be stored. Fourier descriptros only require 3 identifiers, where as complex moments and Zernike moments need 6, Hu moments 7 and psuedo Zernike moments 9.

4.1 Simulated Star Positions

Evaluating the new algorithms purely on their input data, the star positions, is essential to characterizes their performance. Noise is added to the ideal positions through a zero-mean circular Gaussian distribution. Two separate metrics are presented here: the percentage of correctly identified stars 3a and the percentage of identified stars in the image 3b. The first one shows, that, if a star is identified in the image, all techniques have an accuracy above 99.9% under various levels of added noise. However, when the second metric is considered, one sees, that fewer stars in the image get identified with increasing noise. This is relevant, when computing the attitude, as one will see in the next section 4.2. Additionally, there is now a clear distinction between the performances of the different algorithms. The reference method from Liebe identifies the most stars, closely followed by Fourier Descriptors. At higher levels of noise, both fall behind Hu moments and complex moments, and later even Zernike moments.

4.2 Simulated Images

In this test the stars first need to be extracted from the image and their position needs to be determined. Only then, the star identification can be performed and afterwards the attitude is computed. All performance metrics can be seen in Table 2. All six different star identification algorithms achieve similar

Star identification algorithm	**Mean angular deviation [arcsec]**	**Dropped results [%]**	**Average runtime [ms]**
Hu moments	-0.227421 ± 9.42965	0.01	1.35731
Complex moments	-0.227648 ± 9.49849	0.02	1.59177
Zernike moments	-0.237213 ± 11.0216	0.02	1.70367
pseudo Zernike moments	-0.24811 ± 10.0752	0.01	1.94534
Fourier descriptors	-0.229222 ± 9.26181	0.01	1.31561
Liebe	-0.211116 ± 9.01989	0.01	1.3036

Table 2: Performance of the complete system on simulated images for different star identification algorithms.

arcsecond accuracy. However, there are small differences between them. The order is identical to the previous test at low levels of added noise, with Liebe's triangle being the most accurate one, closely followed by Fourier descriptors. Hu moments and complex moments perform almost identically, with Hu moments slightly ahead of complex moments. Pseudo Zernike moments are behind the two, and Zernike moments are the worst in terms of standard deviation. Out of the 10000 tested attitudes, most algorithms only dropped a single image. Only complex moments and Zernike moments dropped two images. The average runtime for the different star identification methods can also be seen in Table 2. They are meassured on an Intel i9-9900K, which is more powerful than the typical processor in a star tracker. Thus, the runtimes only depict relative performance differences between the techniques. Liebe's

triangle method is the fastest, closely followed by Fourier descriptors. Behind them are Hu moments and then complex moments. Zernike moments are the slowest, with the pseudo Zernike moments being slower than the classical Zernike moments.

4.3 Real Images

To verify the performance of the star tracker not only on simulated images, real images were also used. A total of 63 images covering most parts of the Northern Hemisphere with arbitrary rotations were captured using a camera. The reference attitude for each image was determined using *Astrometry.net* [26]. The results of it can be seen in Table 3. In contrast to the previous test, Hu moments, complex moments and pseudo Zernike moments have a higher accuracy than Fourier descriptors. Only in terms of failed identifications do Fourier descriptors have a slight advantage over the other techniques. Zernike moments perform the worst, and Liebe's triangle the best in both metrics. The magnitude of the deviation is significantly higher than in the simulated test. This has various reasons: It is unknown how accurate the reference attitude determined by *Astrometry.net* is and how large the error is. Ground-based observations are always influenced by the atmosphere. It dims the light from the stars and adds disturbances to the image through the motion of the air. Furthermore, there is the problem of light pollution caused by artificial lights of nearby cities which limits the visibility of the stars further. All effects also change with the cameras orientation and how close its FOV is to the horizon. Taking all these effects into account, the performances of the different algorithms are within the expected range.

Star identification algorithm	**Mean angular deviation [deg]**	**Failed identifications [%]**
Hu moments	-0.0522763 ± 0.101105	4.76
Complex moments	-0.0517002 ± 0.101513	4.76
Zernike moments	-0.162016 ± 0.728742	17.46
pseudo Zernike moments	-0.0573003 ± 0.0995736	7.94
Fourier descriptors	-0.11398 ± 0.388429	1.59
Liebe	-0.0577865 ± 0.0986445	1.59

Table 3: Performance of the complete system on real images for different star identification algorithms.

5 Conclusion

This work showed that Hu moments, complex moments, (pseudo) Zernike moments and Fourier descriptors can be used for star identification in a star tracker. First it was discussed how a suitable pattern can be created from the stars in the image and what criterions should be met. After fitting patterns were created, it is shown how Hu moments can be used to identify these patterns. The same concepts are then also applied to complex moments, Zernike moments and Fourier descriptors. For the Fourier descriptors, a variation of the star pattern is created. All algorithms achieve similar accuracies, but they differ in how many stars are correctly identified. Fourier descriptors fall shortly behind the reference algorithm in terms of accuracy, and match it in speed and memory requirements. Hu moments and complex moments perform slightly worse than Fourier descriptors in every metric. They also come with higher memory requirements. Compared to the similar complex moments, Hu moments always have a minute advantage. Zernike moments can use different radial polynomials. In their original form the accuracy is the worst out of all presented methods. This can be improved by using the pseudo radial polynomial, however, more identifiers are necessary to achieve this, which increases processing time and storage requirements. Regardless of the used radial polynomial, Zernike moments always fall behind the other methods. The lead of Liebe's triangles and Fourier descriptors is lost at higher levels of position noise. All methods were also able to accurately identify the stars in real images. While none of the newly presented techniques offer significant advantages compared to the already established algorithms, the application of knowledge from the field of pattern recognition to star trackers is an interesting idea worthwhile investigating.

References

[1] Carl Christian Liebe. "Pattern recognition of star constellations for spacecraft applications". In: *IEEE Aerospace and Electronic Systems Magazine* 7.6 (1992), pp. 34–41.

[2] Carl Christian Liebe. “Star trackers for attitude determination”. In: *IEEE Aerospace and Electronic Systems Magazine* 10.6 (1995), pp. 10–16.
[3] Benjamin B Spratling IV and Daniele Mortari. “A survey on star identification algorithms”. In: *Algorithms* 2.1 (2009), pp. 93–107.
[4] David Rijlaarsdam et al. “A survey of lost-in-space star identification algorithms since 2009”. In: *Sensors* 20.9 (2020), p. 2579.
[5] Simon Conseil, Salah Bourennane, and Lionel Martin. “Comparison of Fourier descriptors and Hu moments for hand posture recognition”. In: *2007 15th European Signal Processing Conference*. IEEE. 2007, pp. 1960–1964.
[6] Fu Yan, Wang Mei, and Zhang Chunqin. “SAR image target recognition based on Hu invariant moments and SVM”. In: *2009 Fifth International Conference on Information Assurance and Security*. Vol. 1. IEEE. 2009, pp. 585–588.
[7] ALC Barczak et al. “A new 2D static hand gesture colour image dataset for ASL gestures”. In: (2011).
[8] Curtis Padgett and Kenneth Kreutz-Delgado. “A grid algorithm for autonomous star identification”. In: *IEEE Transactions on Aerospace and Electronic Systems* 33.1 (1997), pp. 202–213.
[9] Meng Na, Danian Zheng, and Peifa Jia. “Modified grid algorithm for noisy all-sky autonomous star identification”. In: *IEEE Transactions on Aerospace and Electronic Systems* 45.2 (2009), pp. 516–522.
[10] Daniele Mortari. “Search-less algorithm for star pattern recognition”. In: *The Journal of the Astronautical Sciences* 45.2 (1997), pp. 179–194.
[11] Daniele Mortari et al. “The pyramid star identification technique”. In: *Navigation* 51.3 (2004), pp. 171–183.
[12] Enrico Silani and Marco Lovera. “Star identification algorithms: Novel approach & comparison study”. In: *IEEE Transactions on Aerospace and Electronic systems* 42.4 (2006), pp. 1275–1288.
[13] Ming-Kuei Hu. “Visual pattern recognition by moment invariants”. In: *IRE transactions on information theory* 8.2 (1962), pp. 179–187.
[14] KC Otiniano-Rodrıguez, G Cámara-Chávez, and D Menotti. “Hu and Zernike moments for sign language recognition”. In: *Proceedings of international conference on image processing, computer vision, and pattern recognition*. 2012, pp. 1–5.
[15] M Chandrajit, R Girisha, and T Vasudev. “Multiple objects tracking in surveillance video using color and hu moments”. In: *Signal & Image Processing: An International Journal (SIPIJ)* 7.3 (2016), pp. 16–27.
[16] Jan Flusser. “On the independence of rotation moment invariants”. In: *Pattern recognition* 33.9 (2000), pp. 1405–1410.
[17] Michael Schlemmer et al. “Moment invariants for the analysis of 2D flow fields”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1743–1750.
[18] Michael Reed Teague. “Image analysis via the general theory of moments”. In: *Josa* 70.8 (1980), pp. 920–930.
[19] Zernike von F. “Beugungstheorie des schneidenver-fahrens und seiner verbesserten form, der phasenkontrastmethode”. In: *physica* 1.7-12 (1934), pp. 689–704.
[20] Amir Tahmasbi, Fatemeh Saki, and Shahriar B Shokouhi. “Classification of benign and malignant masses based on Zernike moments”. In: *Computers in biology and medicine* 41.8 (2011), pp. 726–735.
[21] Fatemeh Saki et al. “Fast opposite weight learning rules with application in breast cancer diagnosis”. In: *Computers in biology and medicine* 43.1 (2013), pp. 32–41.
[22] AB Bhatia and E Wolf. “On the circle polynomials of Zernike and related orthogonal sets”. In: *Mathematical Proceedings of the Cambridge Philosophical Society*. Vol. 50. 1. Cambridge University Press. 1954, pp. 40–48.
[23] RL Cosgriff. “Identification of shape, Ohio State Univ”. In: *Res. Foundation, Columbus, Rep* (1960), pp. 820–11.
[24] R Diaz De Leon and Luis Enrique Sucar. “Human silhouette recognition with Fourier descriptors”. In: *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*. Vol. 3. IEEE. 2000, pp. 709–712.
[25] Ali Afzali, Farshid Babapour Mofrad, and Majid Pouladian. “Contour-based lung shape analysis in order to tuberculosis detection: modeling and feature description”. In: *Medical & biological engineering & computing* 58.9 (2020), pp. 1965–1986.
[26] D. Lang et al. “Astrometry.net: Blind astrometric calibration of arbitrary astronomical images”. In: *AJ* 137 (2010). arXiv:0910.2233, pp. 1782–2800.