Deep-sky image live stacking via star descriptor

Haiyang Zhou (周海洋)[1,*] and Yunzhi Yu (余韵致)[2]

[1]*State Key Laboratory of Modern Optical Instrumentation, College of Optical Science and Engineering, Zhejiang University, Hangzhou 310027, China*

[2]*Computer Science Department, University of Southern California, 941 Bloom Walk, Los Angeles, CA 90089, USA*

**Corresponding author: hyayh@zju.edu.cn*

Received July 26, 2016; accepted October 21, 2016; posted online November 16, 2016

Image registration is an old topic but has a new application in deep-sky imaging fields named live stacking. In this Letter, we propose a live stacking algorithm based on star detection, description, and matching. A thresholding method based on Otsu and centralization is proposed to implement star detection. Then, a translation and rotation invariant descriptor is proposed to provide accurate feature matching. Extensive experiments illustrate that our proposed method is feasible in deep-sky image live stacking.

OCIS codes: 150.1135, 100.4994, 350.1260.

doi: 10.3788/COL201614.121501.

Consumer astronomy photography involves deep-sky and planetary imaging. For deep-sky imaging, long exposure times are necessary because of the dim light from the scene. However, the increased exposure time will bring more hot noise and distortion. The alternative plan to this situation is image stacking, an old topic but one that has a novel application in deep-sky imaging fields. There are two methods of live stacking: average and additive stacking. For average stacking, the goal is to improve the signal-to-noise ratio, while additive stacking is designed to achieve a preview result that used to be from long exposures but can instead be achieved with a relatively short exposure time. There is already some offline deep-sky stacking software, such as DeepskyStacker[1], Registax[2], and PixInsight[3]. Since they are all offline-use software products, the user cannot preview the stacked result while photographing. Recently, SharpCap[4] provided a new function named live stacking, which stacks multiple frames and allows users to preview the result immediately. However, it is still a highly parameter-related application. No matter which method of stacking is implemented in the above-mentioned software, accurate registration between images is always a prerequisite of stacking.

Image registration is the process of overlaying two or more images of the same scene taken at different times[5]. It is required in many fields, including remote sensing, laser radar and aerial optical imagery[6,7], multi-focus, microscopic, and fluoroscopic image registration[8–11], optical stabilization[12], and other 2D and 3D applications[13–16]. Image registration in deep-sky image live stacking mainly handles translation and rotation deformations caused by the vibrations of imaging devices and the Earth's rotation.

As a kind of image registration method, feature-based image registration is the current research hotspot; we refer the reader some articles for an overview of this method[5,17,18]. A standard feature-based image registration method consists of feature detection, feature description, feature matching, transformation estimation, and image resampling. Among these parts, feature detection, feature description, and feature matching are the critical parts whose accuracy will affect the final registration result. Compared with edges, corners, endpoints, intersections, and other features, stars serve as relatively ideal features for deep-sky images.

The most commonly used star detection method is the thresholding method, where every pixel that fulfills a certain intensity, area, or distance criterion will be considered as star pixel. A few thresholding-related star detection algorithms have been developed in the past few years. Cristo *et al.*[19] proposed a novel thresholding method for detecting stars automatically. That method introduced a set of image pre-processing and post-processing models, so it is not simple enough to be applied in live situations. Wang *et al.*[20] proposed a fast onboard star extraction algorithm, which is more like a bright star extraction algorithm and cannot be applied in our dim deep-sky image stacking applications. Arbabmir *et al.*[21] proposed a thresholding method that performs image binalization properly even for images with uneven illumination. Regardless of its iterative nature, its performance still depends on the size of the local window, which additionally depends on the optics design of the camera, and that limits its popularity. Xu *et al.*[22] described a new weighted threshold algorithm based on the estimation of the optimal threshold for achieving minimal centroid error. As an iterative method, the computation efficiency is difficult to guarantee without the proper initial threshold. However, consumer software, such as Registax, SharpCap, and PixInsight, still requires user to input the threshold manually. Although PixInsight implemented a wavelet-based automatic star detection algorithm, its detection criteria are still controlled through process parameters. Nowadays, these methods suffer either a computation or parameter-dependent problem. In this Letter, an automatic and efficient thresholding technique based on Otsu and centralization is proposed. It could process deep-sky images even with existing illumination differences. In the proposed method, the threshold firstly is calculated by

Otsu and then centralized to make it feasible for uneven intensities.

In order to match features between different images, a feature descriptor is necessary. In computer vision fields, several feature descriptors, such as SIFT, SURF, KAZE, etc., have been proposed for complicated computer vision applications. The rotation invariant property of these descriptors either comes from the local statistics of gradients or Haar wavelet responses. However, for star features used in deep-sky images, this rotation invariant nature may fail due to the star's special shape, where the intensity decreases similarly in all directions. In order to handle this problem and make the star feature rotation invariant, we creatively proposed a translation and rotation invariant descriptor. This descriptor relies on the star's characteristics and relationship with its nearby stars. The best candidate star from the deformed image for each star from the reference image can be efficiently obtained by calculating the Euclidean distance.

Figure 1 shows the flow diagram of our proposed algorithm. Besides the pre-processing and post-processing steps, our approach involves three core components: star detection, star description, and star matching. Since our algorithm is designed to perform live stacking, after pre-processing, such as subtracting the dark, bias, and flat-field frames, each frame will be sent to the stack thread, which works with the grab thread asynchronously. Then the stack thread will register the input frame to the reference frame one by one, and the final stacked result will be used as the new reference frame.

First, stars should be extracted as accurately as possible and in large amounts. Otherwise, the following steps will produce errors and even cause registration failure. In this Letter, we proposed a thresholding method based on Otsu and centralization. The famous Otsu[23] method thresholds a given image by maximizing the inter-class variance. Given an image with L gray levels and N total pixels, the inter-class variance for the chosen threshold t can be calculated by

$$\sigma(t) = w_A(\mu - \mu_A)^2 + w_B(\mu - \mu_B)^2. \tag{1}$$

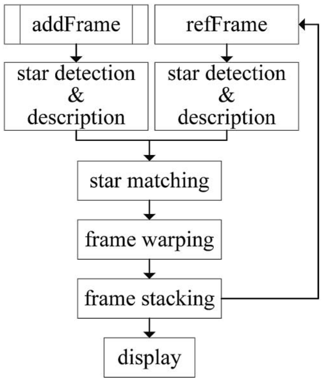

Fig. 1. Flow diagram of our proposed algorithm.

The means μ, μ_A, and μ_B are defined as

$$\mu = \sum_{i=1}^{L} P_i * i, \quad \mu_A = \frac{\sum_{i=1}^{t} i \cdot P_i}{w_A}, \quad \mu_B = \frac{\sum_{i=t}^{L} i \cdot P_i}{w_B}, \tag{2}$$

and the probabilities P_i, w_A, and w_B are defined as

$$P_i = \frac{N_i}{N}, \qquad w_A = \sum_{i=1}^{t} P_i, \qquad w_B = \sum_{i=t}^{L} P_i. \tag{3}$$

The main drawback of Otsu is that it may fail for deep-sky images with uneven illuminations; this commonly takes place when the imaging device is exposed to undesirable light sources. This is a common problem that all global thresholding methods face. We modify the Otsu threshold by introducing centralization, and that makes it applicable to uneven background intensities and noise. This improvement can be achieved by

$$T_c = M_f - t_{\text{mean}} + t_{\text{otsu}}, \tag{4}$$

where M_f represents the mean filtered image matrix, t_{mean} denotes the mean value of the filtered image, and t_{otsu} is the global threshold Otsu method identified. T_c is the centralized pixel-related threshold matrix, which includes thresholds for every pixel in a given image. Then, the binary image can be easily achieved by applying the threshold matrix to the input frame. Figure 2 shows a thresholding example using M42 image. As shown in Fig. 2, compared with Otsu, our result achieves higher detection accuracy, even for these nebula regions.

For stars, the feature is more like a speckle instead of a point. Therefore, we need to extract the stars' region before proceeding further. In order to extract the region of stars, we apply the Moore-Neighbor tracing algorithm modified by Jacob's stopping criteria.

The state-of-the-art algorithms used in deep-sky image stacking only use a star's position and brightness to describe it. This is not sufficient to achieve a rotation invariant nature. In order to keep the registration rotation invariant, they use the triangle similarity to cope with this rotation problem during star matching. So state-of-the-art algorithms are not only heavily reliant on memory to save the formed triangles, but they also have a very expensive computational cost. So, we creatively proposed a descriptor which not only includes a star's characteristics, such as position and brightness, but also takes the spatial relationship with their nearby stars into consideration. This idea of using the geometrical relationship has been proven effective by Shi *et al.*[24], who proposed a topology-based affine invariant descriptor. Topology is constructed among regions, and the number of relative neighboring regions is not fixed. Therefore, its computational cost may increase while processing deep-sky images because that star may have a large number of relative neighboring pairs. Thus, we need to propose a new descriptor used to process deep-sky images. Because of the artifacts of the imaging system and the effects of atmospheric turbulence, the shape of the star mostly looks like an

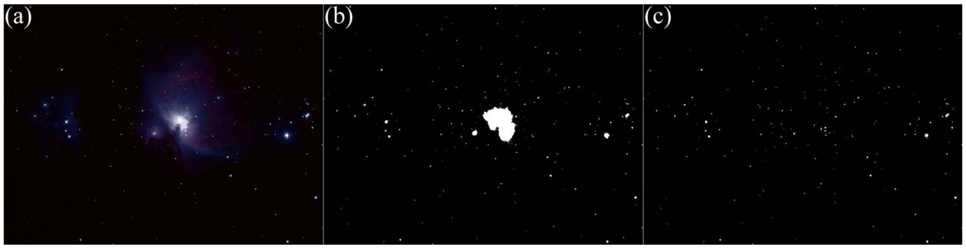

Fig. 2. Comparison of thresholding results: (a) Original M42 image. (b) Otsu result. (c) Our result.

ellipse instead of an ideal circle. In our implementation, since we have extracted the counters of stars, we can simplify the procedure to fit the best ellipse for the counters using the direct ellipse fit method[25]. The ellipse's semi-major axis a, semi-minor axis b, center coordinates (x_c, y_c), and rotation angle θ can be easily obtained using the coefficients of a standard ellipse function.

Then, we introduce two orientations to each star to make it rotation invariant. As shown in Fig. 3, for the current star A, B and C are the two nearest neighboring stars. Connecting A and the two neighboring stars forms two line segments AB and AC, whose lengths are L_1 and L_2, respectively. We mark the angles' line segments AB and AC to the major axis direction x' as θ_1 and θ_2. Finally, we add L_1, θ_1, L_2, and θ_2 to current star's descriptor. Now, our descriptor is formed as

$$\{x_c, y_c, a, b, \theta, L_1, \theta_1, L_2, \theta_2\}. \tag{5}$$

Figure 4 shows our description results for two deep-sky images with minor rotation. The white quiver denotes the orientation, and its length is the distance between two stars. Taking the dominant orientation into consideration, the relationships between the stars are kept perfectly between the two images. Compared with SIFT and SURF, our descriptor is only an 8-vector descriptor, so it is efficient to compute and match by the simplest Euclidean distance. The proposed descriptor also makes our algorithm perform much more efficiently than methods based on triangle similarity.

In the final step of star matching, we applied the random sample consensus (RANSAC)[26] to reject mismatches and calculate the transformation matrix to be used for the post-processing procedures.

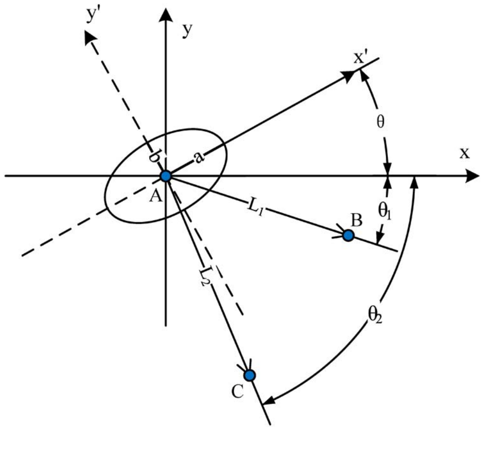

Fig. 3. Schematic diagram of the rotation invariant descriptor.

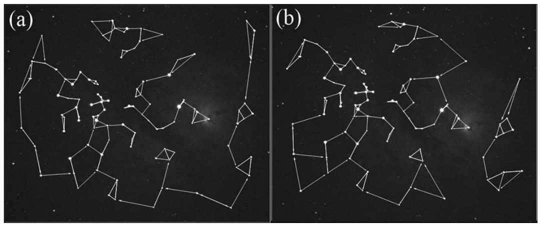

Fig. 4. Description results using our proposed star descriptor.

As mentioned earlier, the mainstream SIFT and SURF cannot be directly applied in our technique, because the dominant orientation obtained by these algorithms is not robust enough to keep the descriptor rotation invariant. This is mainly due to two reasons: the region size used for assigning the dominant orientation and the region size used for the describing the feature. For star features,

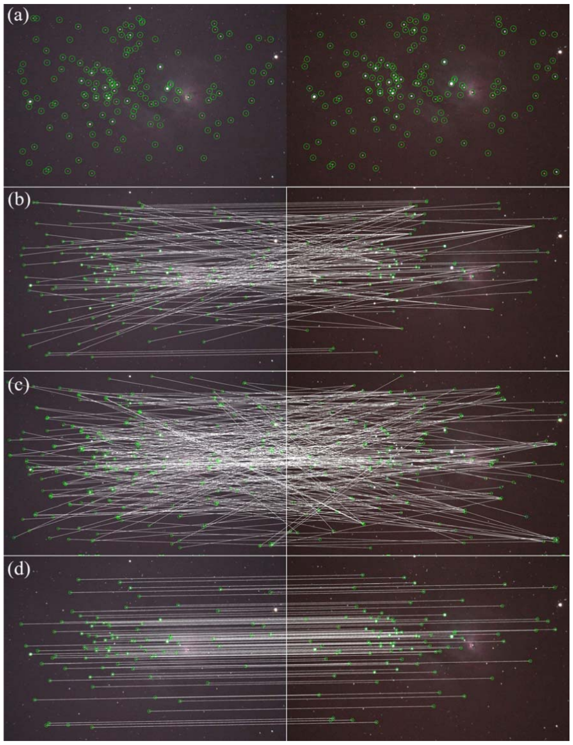

Fig. 5. Matching results using SIFT, SURF, and our descriptor. (a) Star detection results. (b) SIFT result. (c) SURF result. (d) Our result.

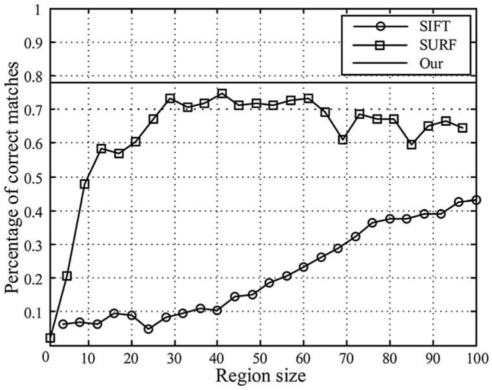

Fig. 6. Matching ratio results among SIFT, SURF, and our descriptor.

different region sizes will result in different matching results. Ruiz-del-Solar *et al.*[27] applied the SIFT descriptor to stellar image matching and introduced an additional verification procedure to discard false detections. With the help of the additional verification procedure, it achieved acceptable results. In other words, regardless of the additional procedure, the defect of the SIFT descriptor still exists.

In order to illustrate the superiority of our descriptor, we compare it with SIFT and SURF using two real deep-sky images captured by a Canon EOS 1100D with a 50 mm lens. For a fair comparison, the features used by the three descriptors are the star centroid points detected by our star detection algorithm. Figure 5 shows the results. The region size for SIFT and SURF is the maximum star contour's size. As shown in Fig. 5, because of the rotation invariant descriptor's inaccuracy, both SIFT and SURF generate many mismatches, while our descriptor achieves a sufficient number of correctly matched pairs.

In order to further illustrate the uncertainty that different region sizes bring into feature matching, we plot the percentage of correct matches with different region sizes in Fig. 6. The horizontal axis denotes the region size, while the vertical axis denotes the percentage of correct matches. As shown in Fig. 6, both the SIFT and SURF descriptors vary dramatically with the increase of the region size. Among the three descriptors, our descriptor obtains the best matching result. Although, compared with SIFT, SURF finally achieves a relatively good and stable result that nearly reaches our result, it still needs expert experience to achieve better results.

Evaluating the registration quality and accuracy is a necessary part of image registration because, without quantitative evaluation, no registration method can be accepted for practical use[5]. Therefore, we do an evaluation on ten randomly rotated (1°–90°) and translated (1–10 pixels) images by root-mean-squared error (RMSE) and joint entropy[28] against SIFT and SURF. The results are shown in Table 1. For both measures, the smaller the calculated value is, the better the registration will be. This quantitative evaluation also could facilitate the following analysis of the influence of the registration accuracy on image live stacking. As shown in Table 1, our descriptor obtains a smaller RMSE and joint entropy than SIFT and SURF overall. This result represents that our descriptor has a higher registration accuracy. For offline registration or the situation where there is a small amount of images to be registered, SIFT and SURF could achieve acceptable results. However, in deep-sky image live stacking, every intermediate averaged or stacked image will be assigned as the new reference frame used for the next registration. The error produced by the inaccurate registration will be accumulated after averaging or stacking images continuously and finally leads to the registration failure. So, a higher registration accuracy is required.

In our final experiment, where the preview resolution is 1920 × 1200, the computation cost for one frame is kept below 0.5 s, and this is acceptable in deep-sky live stacking. Although the efficiency already is high enough to ensure a live process, we still introduce a buffering mechanism in the implementation to avoid the problem that the calculation cannot keep up with the frame rate under a very low exposure. Every grabbed frame will be saved in the buffering pool while the stack thread picks frames from the buffering pool one by one. These two threads work with each other asynchronously. If the frame rate is high and the calculation of the image registration cannot catch up with it, the old frames saved in the buffering pool will be dropped and the newly arrived frames will be added in. This implementation always guarantees the latest frame will be stacked into the preview result.

Table 1. Quantitative Registration Evaluation Against SIFT and SURF

	RMSE									
SIFT	0.8880	0.4051	0.0161	0.1150	0.7101	0.8650	0.4309	0.9072	0.4314	0.9771
SURF	0.1475	0.2306	0.0152	0.0704	0.4595	0.7986	0.0199	0.5854	0.3452	0.7318
OUR	0.0106	0.0180	0.0143	0.0455	0.0224	0.0281	0.0127	0.0380	0.0307	0.0230
	Joint Entropy									
SIFT	9.3339	9.3715	8.3369	9.3600	9.4503	9.3734	7.0749	9.3636	9.3717	9.8534
SURF	9.3354	9.3711	7.3365	9.3588	9.3185	9.3651	6.2628	9.3414	9.3663	7.2840
OUR	6.0073	6.2474	6.6522	8.0960	6.8046	7.0414	6.1535	7.0749	7.4858	6.8305

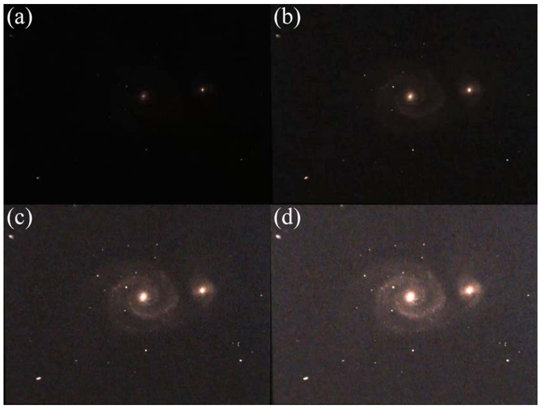

Fig. 7. Live stacking results while photographing M51. (a) No stack. (b) Three frames stacked. (c) Six frames stacked. (d) Nine frames stacked.

Figure 7 shows the live stacking results of M51. These M51 images are captured with a 15-s exposure and a 10 gain camera setting. Figure 7(a) is the original frame without stacking any more images. Due to the dim light, the original frame shows nothing except several bright points. With the increase of the stacked image number, the details of M51 emerge gradually without any affecting the translation and rotation. The black edges in Fig. 7(d) illustrate that translation and rotation between frames do exist. This is caused by setting pixels that does not exist in the current frame compared with the reference frame to zero.

Figure 8 shows two ten-frames-stacked M92 images where every frame is captured using 5 s of exposure and a 10 gain camera setting with or without alignment enabled. Both ten-frames-stacked results in Fig. 8 make more stars emerge and achieve a long-exposure effect. However, in Fig. 8(b), there are clear star trails introduced by the equatorial mount's inaccurate compensation for the Earth's rotation, while the stars' shapes are still sharp in Fig. 8(a). That means even when using a short exposure time, image registration is still needed, and our live stacking method accurately registers every input frame and successfully eliminates these inaccurate compensations.

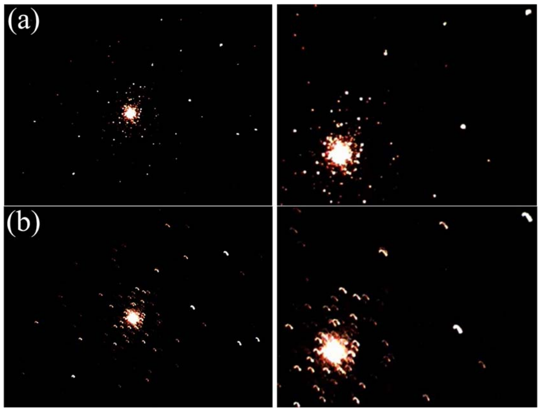

Fig. 8. Comparison results using ten-frames-stacked M92 images and their zoomed views. (a) Result with alignment enabled. (b) Result with alignment disabled.

In conclusion, we propose a live stacking algorithm based on star detection, description and matching. Our star detection overcomes the problem that the traditional global threshold-based method using centralization. The major contribution of our work is the proposed descriptor, which makes for up the defect of traditional descriptors, which cannot provide image registration in deep-sky imaging. As a new descriptor, our star descriptor has fewer memory requirements and is better for deep-sky image live stacking. The experiments indicate that our proposed live stacking algorithm achieves a long-exposure effect with translation and rotation invariant features.

References

1. L. Coiffier, "DeepSkyStacker," http://deepskystacker.free.fr/english/ (2016).
2. C. Berrevoets, "RegiStax," http://www.astronomie.be/registax/ (2016).
3. J. Conejero, "PixInsight," https://pixinsight.com/ (2016).
4. G. Robin, "SharpCap," http://www.sharpcap.co.uk/ (2016).
5. B. Zitová and J. Flusser, Image Vision Comput. **21**, 977 (2003).
6. J. Chen, H. Feng, K. Pan, Z. Xu, and Q. Li, Optik-Int. J. Light Electron Opt. **125**, 697 (2014).
7. A. Mastin, J. Kepner, and J. Fisher, in *IEEE Conference on Computer Vision and Pattern Recognition, 2009* (CVPR, 2009), 2639.
8. C.-W. Wang, S.-M. Ka, and A. Chen, Sci. Rep. **4**, 6050 (2014).
9. Y. Liu and F. Yu, Opt. Commun. **341**, 101 (2015).
10. W. Bingjian, L. Quan, L. Yapeng, L. Fan, B. Liping, L. Gang, and L. Rui, Appl. Opt. **50**, 1861 (2011).
11. S. Liu and H. Hua, Opt. Express **19**, 353 (2011).
12. Q. Yang, J. Zhang, K. Nozato, K. Saito, D. R. Williams, A. Roorda, and E. A. Rossi, Biomed. Opt. Express **5**, 3174 (2014).
13. M. Pohit and J. Sharma, Appl. Opt. **54**, 4514 (2015).
14. X. Dong, Y. Zheng, S. Bai, W. Xu, and X. Huang, Chin. Opt. Lett. **12**, 121002 (2014).
15. P. Cao, Y. Yang, C. Li, H. Chai, Y. Li, S. Xie, and D. Liu, Chin. Opt. Lett. **13**, 041102 (2015).
16. X. Fan, C. Zhou, S. Wang, C. Li, and B. Yang, Chin. Opt. Lett. **14**, 081101 (2016).
17. T. Tuytelaars and K. Mikolajczyk, Found. Trends. Comput. Graph. Vision **3**, 177 (2008).
18. R. Szeliski, Found. Trends. Comput. Graph. Vision **2**, 1 (2006).
19. A. Cristo, A. Plaza, and D. Valencia, in *IEEE International Symposium on Signal Processing and Information Technology* (ISSPIT, 2008), 180.
20. T. Wang, Y. Qiu, H. Cai, and J. Deng, Sci. China Phys. Mech. Astron. **53**, 51 (2010).
21. M. V. Arbabmir, S. M. Mohammadi, S. Salahshour, and F. Somayehee, J. Opt. Soc. Am. A **31**, 794 (2014).
22. W. Xu, Q. Li, H.-J. Feng, Z.-H. Xu, and Y.-T. Chen, Optik-Int. J. Light Electron Opt. **124**, 4673 (2013).
23. N. Otsu, Automatica **11**, 23 (1975).
24. C. Shi, G. Wang, X. Lin, Y. Wang, C. Liao, and Q. Miao, in *2010 IEEE International Conference on Image Processing* (2010), p. 133.
25. A. Fitzgibbon, M. Pilu, and R. B. Fisher, IEEE Trans. Pattern Anal. Mach. Intell. **21**, 476 (1999).
26. M. A. Fischler and R. C. Bolles, Commun. ACM **24**, 381 (1981).
27. J. Ruiz-del-Solar, P. Loncomilla, and P. Zorzi, in *Progress in Pattern Recognition, Image Analysis and Applications: 13th Iberoamerican Congress on Pattern Recognition, CIARP 2008, Proceedings*, J. Ruiz-Shulcloper and W. G. Kropatsch, eds. Havana, Cuba, Sept. 9–12, 2008 (Springer, 2008), p. 618.
28. A. A. Goshtasby, *Image Registration: Principles, Tools and Methods* (Springer Science & Business Media, 2012).