CENTROID ESTIMATION BASED ON MSER DETECTION AND GAUSSIAN MIXTURE MODEL

Wangbin Ding, Dong Gong, Yanning Zhang and Yao He

School of Computer Science, Northwestern Polytechnical University, Xi'an, China
dingwangbin@126.com, edgong01@gmail.com, ynzhang@nwpu.edu.cn, yaoh.nwpu@gmail.com

ABSTRACT

This paper proposes a sub-pixel centroid estimation algorithm for star field image recorded by CCD sensor. To handle the connected as well as partially overlapped star spots in a star region, we involve the the MSER (Maximally Stable Extremal Region) detector to discover the star spots and related characteristics of latent star spots in a star region. After extracting the star regions, we will estimate the locations of centroids in each star region simultaneously using the GMM (Gaussian Mixture Model) and Expectation-maximization (EM) algorithm, instead of segment each star spots in a star region. Using our algorithm, overlapped star spots can be discovered easily and their centroids can also be estimated effectively. Experiments on simulation data sets demonstrate effectiveness and robustness of this algorithm.

***Index Terms*—** Star field image processing, sub-pixel centroid estimation, MSER, GMM.

1. INTRODUCTION

High-precision sub-pixel location estimation of star centroids in star field image is a key aspect for astro observation, spacecraft-attitude control and celestial navigation. Sub-pixel centroid estimation for the star regions is usually a non-trivial topic, as centroid estimation accuracy are usually influenced by discrete sampling of the CCD, severe photon noise caused by guide star, readout noise of the CCD, speckle noise introduced by the atmosphere [1] and arbitrary pattern of star target. Over the past two decades, centroid estimation algorithms have been investigated widely by many researchers. Two main techniques for the centroid estimation are: (1) interpolation-based techniques, and (2) distribution or fitting-based techniques. For the interpolation based techniques, Sun et. al. proposed a centroid method based on energy accumulation [2]. Brendan M. Quine et. al. developed a theoretical methodology to estimate the location of star centroid [3]. By using interpolation they are enable to estimate the location of star centroid without extensive characterizations that required to derive fitting coefficients. For the fitting-based techniques, Shen and Jiang used a Gaussian model to fit the profile of focal spot, the center of the fitting function implies the centroid of the star region [4]. In addition, S. Tomas et. al. proposed another three main kinds of centroid algorithms: quad cell (QC) estimator, centre of gravity(CoG) approaches and correlation (Corr) methods to boost the centroid estimation performance [1].

All of these techniques focus on estimating a centroid from a star region with only one star spot. However, in real-world application, star image patterns in a star field image are usually arbitrary. Spots of adjacent stars often connect or partially overlap. Fig. 1 illustrates a typical case that two closely-located star spots are needed to be separated and their centroids are needed to be estimated. We separate the task of centroid estimation under star spot connecting condition into two main sub-tasks: (1) discovering star spots located in one small region and (2) estimation of the related centroids of star spots closely located in one small region.

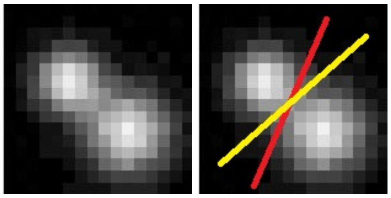


Fig. 1: Overlapping star spots in one samll region bring a great challenge for generating a proper cutting line for star sopt extraction and further task of centroid estimation.

Intuitively, as shown in Fig. 1, pixels located in the region are ambiguous, but locations and intensities of these pixels have an explicit pattern with concentration and diffusion characteristics. We model these locations statistically as samples drawn from a GMM. And the parameters of the GMM can be estimated using EM algorithm [5]. We initialize the parameters of GMM in advance. Especially, the number of components in each region is the key for utilizing the GMM for a real-world star field image. Considering the concentration of the pixels belong to each component, we discover the star spots and initialize the GMM parameters utilizing MSER tree in our algorithm [6, 7].

2. PROPOSED METHOD

Considering the characteristics of star field image and the star regions above, we propose an algorithm to extract candidate star regions, discover star spots and estimate centroids of star spots.

2.1. Candidate Regions Extraction

All star regions and star spots can be described as clusters of stable MSERs, so we can discover and extract all star regions and spots through detecting all MSERs in an star image. Although the MSERs can be detected in a quasi-linear time [8], it is still a great expenses to detect the extremal regions from a whole star image. We improve computation efficiency by extracting candidate start regions from star image in advance. Then, we can detect MSERs from the small candidate star regions instead of the whole star image.

Let $\mathbf{X} \subset \mathbf{R}^2$ be a domain on which a gray-scale image $I : \mathbf{X} \rightarrow [0, 255]$ is defined. Sorting all the pixels $\mathbf{x} \in \mathbf{X}$ with regard to their intensity $I(\mathbf{x})$. Throughout this paper, we regard locations of a pixel as a two dimensional point. By removing 30% pixels with highest intensity and 20% pixel with lowest intensity from $\mathbf{X}$, we can estimate the mean and variance of the star image background from remained pixels $\mathbf{X}'$ robustly. Then the image I can be converted into a binary map I' using Equ. 1,

$$I'(\mathbf{x}) = \begin{cases} 0 & I(\mathbf{x}) < t \\ 1 & I(\mathbf{x}) \geq t \end{cases} \quad (\mathbf{x} \in \mathbf{X}, t = u + 3\sigma) \tag{1}$$

where I' represents the binary map of I, t denotes binarization threshold, u and σ are mean and variance of $\mathbf{X}$ respectively and can be calculated using:

$$u = \frac{1}{n} \sum I(\mathbf{x}_i) \quad (\mathbf{x}_i \in \mathbf{X}') \tag{2}$$

$$\sigma = \frac{1}{n} \sum (u - x_i)^2 \quad (\mathbf{x}_i \in \mathbf{X}^{'}) \tag{3}$$

The candidate regions in star field image I is constructed by pixels $\mathbf{x} \in \mathbf{X}$, whose $I'(\mathbf{x})$ equals to 1. Fig. 2 illustrates a brief process of the candidate regions extraction.

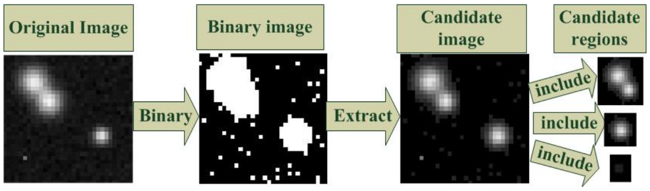


Fig. 2: Relying on the binary map, several disconnected candidate star regions can be extracted from a star field image.

2.2. MSER Tree Generation

The MSER detector provides us a robust feature detection algorithm to detect the MSERs. A region is extremal if the intensities of pixels inside of it are higher than the intensities of those at its boundary. Considering the binary maps generated using Equ. (1) with continuous decreasing threshold t, we indicate $I_{>t} = \{\mathbf{x} | I(\mathbf{x}) > t\}$ as a set of pixel, in which the intensities of pixels are larger than threshold t. As shown in Fig. 3, cardinality of $I_{>t}$ tends to increase as the t decreases. This result is natural because with the decreasing of t, more pixels satisfy the condition in Equ. (1) to be decided as pixels belong to star regions. The stability of a region is calculated based on the variation of the cardinality with regard to the threshold t [9]. Thus, the stability of a region can be defined mathematically as:

$$s(t, d) = \frac{|I_{>(t-d)}| - |I_{>(t+d)}|}{|I_{>t}|} \tag{4}$$

where d in is one of two parameters need to be set in the MSER detector and $| \cdot |$ is the cardinality calculating operation. The other parameter of MSER detector is t_{mser}, stability threshold for labelling MSER from candidate regions. d and t_{mser} will introduced below.

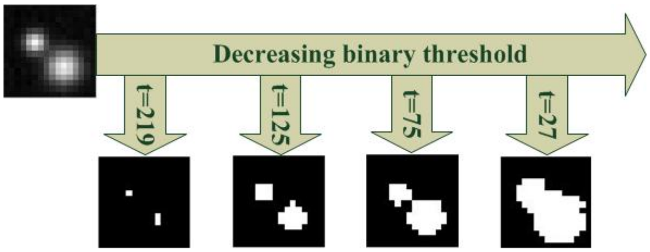


Fig. 3: The white area of the image increased as the threshold decreased. This implies the cardinality of $I_{>t}$ is increased as the threshold decreased.

For each candidate region we can build a correspondent component tree to detect MSERs from it. Vedaldi [10] produced an efficient implementation of MSER detector. In order to build a component tree for a candidate region, we need to organize the pixels of the candidate region into a tree structure based on the intensities of pixels in the region. Fig. 4 illustrates how pixels in a typical star region are organized using a component tree. As mentioned before, the component tree could be built in a quasi-linear time.

After building the component tree from a candidate region, the corresponding MSERs could be computed by travelling the component tree. Each sub tree of the component tree is a candidate MSER that involves a set of topologically connected pixels. We label the sub trees whose stability are less than t_{mser} as MSERs. With the MSERs being labelled, we link the MSER regions to build a MSER tree. As shown in Fig. 5, the MSER tree represents the structures of star spot in one star region.

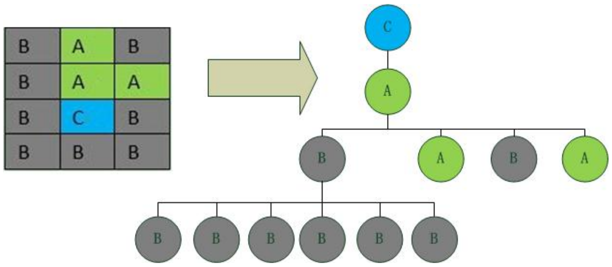


Fig. 4: Suppose that $I(C) < I(A) < I(B)$, we construct the component tree by linking the pixels with high intensity to pixels with low intensity. Thus, in this tree structure, pixels with high intensity will be the leaves and pixels low intensity will be the roots. We must assure that pixels could only be directly linked with each other when they are topologically connected.

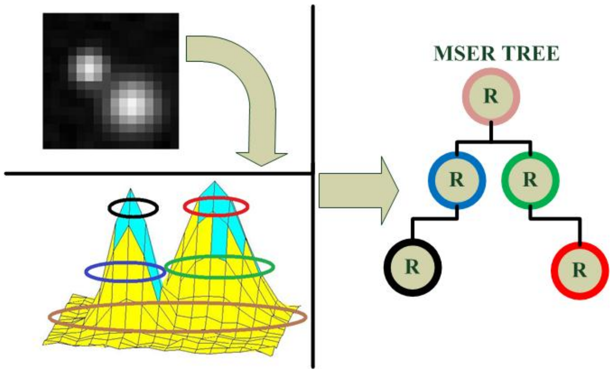


Fig. 5: After detecting the MSERs from the component tree, we will organize the MSERs into a tree structure based on their inclusion relationships. If MSER R2 include MSER R1, then the R1 will be a leaf node of R2 in MSER tree structure.

2.3. Centroid Estimation

2.3.1. GMM Parameter initialization

We model one star spot as a point source with Gaussian distribution incidenting on the surface of a camera sensor e.g. a charge coupled device (CCD) sensor [3]. As we have mentioned before, for the candidate regions that involve more than one star spots, we would like to fit the whole region directly rather than segment it into different parts or extract start spots directly.

Let C be a pixel set corresponding to the largest MSER of a candidate region, and $\mathbf{x}$ is a pixel in C. The intensity of a pixel $I(\mathbf{x})$ is interpreted as the accumulation of photons at the location of pixel $\mathbf{x}$, a.k.a. the point with coordinate $\mathbf{x}$ in the 2-dimensional grid of image. Thus, the two-dimensional GMM model used to fit the distribution of the pixels can be defined as:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \mathbf{\Sigma}_k) \tag{5}$$

where K represents the number of Gaussian component in the GMM, μ_k stands for the mean of k-th Gaussian component, $\mathbf{\Sigma}_k$ is the covariance matrix of k-th Gaussian component and the $\mathbf{x}$ denotes a pixel in the star region. The model in Equ. (5) represents a two-dimensional Gaussian-mixture-shape curve surface with means μs corresponding to centroids of star spots. Finally, the task of identifying the centroids of each star spot in a star region can be converted into estimating the mean of each Gaussian component in GMM.

We use the EM algorithm to estimate the μ_k, $\mathbf{\Sigma}_k$ and π_k in Equ. (5) [11]. As the EM algorithm is an iterative method [12], we should initialize the GMM parameters in advance; and the number of components K also should be setted before estimation process. As the intensity can be realized as result of an accumulation of photons, in order to simplify the model, we assume each pixel $\mathbf{x}$ generates more than one same observations with same property as $\mathbf{x}$ for EM. The size of samples is magnified through a simple *resampling* process. We denote the repeated observation samples resampled from pixel $\mathbf{x}_i$ as $S(\mathbf{x}_i)$, than the set of total observations of C after resampling can be represented as:

$$S(C) = \bigcup_{\mathbf{x}_i \in C} S(\mathbf{x}_i) \tag{6}$$

Usually, K-means clustering is employed to initial the EM parameters [13]; each cluster corresponds to each component in GMM. To decide the number of clusters, we utilize the MSER tree and travel it to get the number of leaf nodes. Intuitively, the number of leaf nodes in MSER tree is equal to the number of Gaussian component K in GMM. The initial center of each cluster μ_k is equal to the pixel who has largest intensity in each leaf node. Having decided the number of cluster and their initial centers for the K-means algorithm. The rest of parameter could be initialized sequentially.

Then we initialize the π_k. Suppose the $S(\mathbf{C})$ is a set of observations $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$. For each sample $\mathbf{x}_i \in S(\mathbf{C})$, we assign it to the cluster whose centroid have shortest Euclidean distance with $\mathbf{x}_i$. By assigning each $\mathbf{x}_i$ to one cluster, whole observation set $S(C)$ will be split into K subset. We denote the observations that belong to k-th cluster as $S_k(C)$. The coefficient π_k could be computed as

$$\pi_k = \frac{|S_k(C)|}{|S(C)|} \tag{7}$$

where the operator $|\cdot|$ represents the cardinality of set $\cdot$. Then the covariance matrix of k-th Gaussian component in GMM $\mathbf{\Sigma}_k$ could be initialize as the covariance of the data in observation set $S_k(\mathbf{C})$. This initialization scheme is concluded as:

1. Initialize the number of Gaussian component as the number of leaf nodes of the MSER tree, which can be got through travelling the MSER tree;
2. Initialize the mean of each Gaussian component as the location of the pixel with the highest intensity in each MSER tree leaf node;
3. Assign pixel samples to different clusters according to Euclidean distance; then calculate the Σ_k and π_k.

2.3.2. GMM Parameters estimation

After initialization of the parameters of GMM, the EM algorithm could be operated in sequence. There are two main steps in each iteration of EM algorithm: *estimation* and *maximization* [14]. Here we list a detail description of EM algorithm for estimating the parameters of a GMM.

1. Expectation: calculate the responsibility value that Gaussian component k takes for the observation $\mathbf{x}_n$ using the current parameters value.

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \mathbf{\Sigma}_k)}{\sum\limits_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\mu_j, \mathbf{\Sigma}_j)} \tag{8}$$

where $\gamma(z_{nk})$ denotes the *responsibility* that component k takes for explaining the observation data set [5].

2. Maximization: Re-estimate the parameters using the current responsibility value.

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}) \tag{9}$$

$$\mu_{\mathbf{k}}{}^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n \tag{10}$$

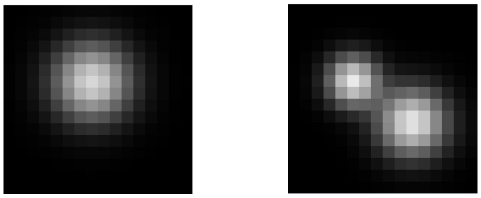
$$\mathbf{\Sigma}_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \mu_k^{new})(\mathbf{x}_n - \mu_k^{new})^{\mathrm{T}} \tag{11}$$

$$\pi_k^{new} = \frac{N_k}{N} \tag{12}$$

The whole process of our method is described in Algorithm 1. First we binarize the star field image to extract candidate star regions from original star field image; and generate the MSER tree from star regions and get the number of leaf nodes as number of components of the latent star spots. Then fitting the largest MSER in candidate star region with a GMM, whose parameters could be estimated using the EM algorithm iteratively. The mean of each component implies the location of the centroid of a star spot in star region.

Algorithm 1: The process of centroid algorithm

```
Input: $I$
Output: $\mu$
$I'$=binarizeImage($I$)
$R$=candaditeRegionExtraction($I$ , $I'$)
foreach $r$ in $R$ do
    $T$=buildMSERTree($r$)
    $S$=generateObservation($r$)
    $P$=initilizeGMMParameter($T$)
    while isConvergence($P$) do
        $P$=Expectation($P$ , $S$)
        $P$=Maximization($P$ , $S$)
return $P \to \mu$
```



(a) A single star region

(b) A double partially overlapped star region

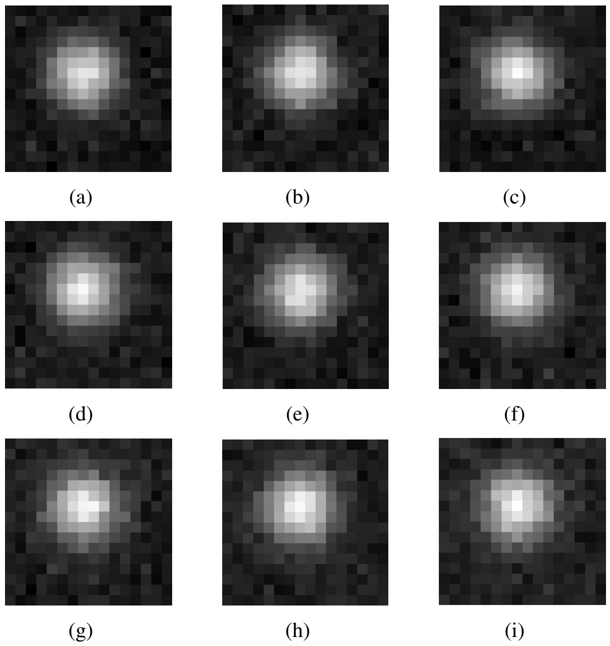
Fig. 6: These two kinds of star images with different Gaussian noises will be used to evaluate our algorithm.

3. EXPERIMENTS

The key principle of our centroid algorithm is that we skirt around the hard task of separating connected or overlapped star spots from a star region. For the star region contains one star spot, we will see later, our method could be as robust as state-of-the-art method. For the star region contains more than one star spots, our method still can estimate the centroid of each star effectively, while other methods fail to handle the conditions with overlapped star spots. Our experiments are conducted on two different types of star image, one of them contains only one star spot(Fig. 6(a)), and the other one contains two star spots(Fig. 6(b)). Different kinds of Gaussian noise are added to this two kinds of image to generate our experiment samples.

In the first experiment, beside our method, other four methods are engaged to estimate the center of star regions. We tested them on 20 individual images. Some of them are listed in Fig. 7. Each of these images is generated by adding different Gaussian noise on image in Fig. 6(a). The mean of the added Gaussian noise is range from 20 to 39 and the variance keep stay in 10. We set the d as 5, and t_{mser} as 1 in this experiment. The process of extracting the largest MSER is shown in Fig. 8. Fig. 9 illustrates the estimation error of each algorithm for each test sample; and the result demonstrates a fact that our method could provide a robust sub-pixel centroid estimation as well as the state-of-the-art method.

In the second experiment, our method is evaluated on the star images that contain a pair of partial-overlapped star spots. Some examples are listed in Fig. 10. Same as the previous experiment, Gaussian noise is added to the star image in Fig. 6(b). The means of the Gaussian noise are ranged from 20 to 39, and the variance keeps as 10. We also set the b as 5, and t_{mser} as 1 in this experiment. The process of extracting a largest MSER is shown in Fig. 11. Fig. 12 shows the error of our algorithm when estimating the centroid location of left star spot in Fig. 6(b); Fig. 13 illustrates the error when estimating the centroid location of right star spot in Fig. 6(b); while other methods fail to estimate the centroid location from such


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 7: Examples of samples that we used to evaluate different algorithms.

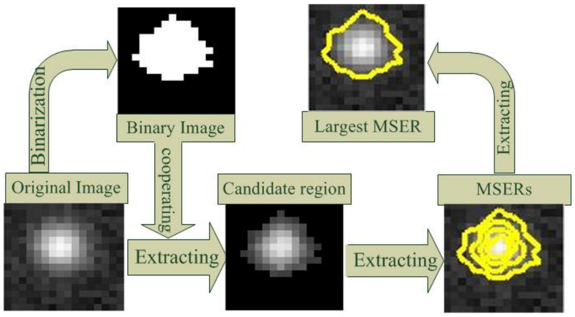


Fig. 8: The process of extracting the largest MSER from star image. Such MSER will be involved to estimate the centroid of start region in our method

kind of data. The results demonstrate that our method can estimate the locations of centroids for the partially overlapped star regions with a high-level sub-pixel accuracy.

4. CONCLUSION

This paper proposes a centroid estimation method based on MSER detection and GMM. We extracted the candidate star regions from the star image in advance. Then, we detect MSERs from the candidate star regions to build the MSER trees. The largest MSER in a candidate star region is the star region we will utilize to calculate the star spot centroid; and by travelling the MSER tree, the parameters of GMM are initialized. Finally, the proposed method estimates the locations of centroids using EM algorithm robustly and precisely.

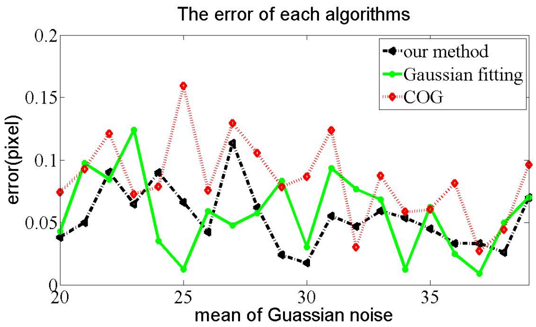
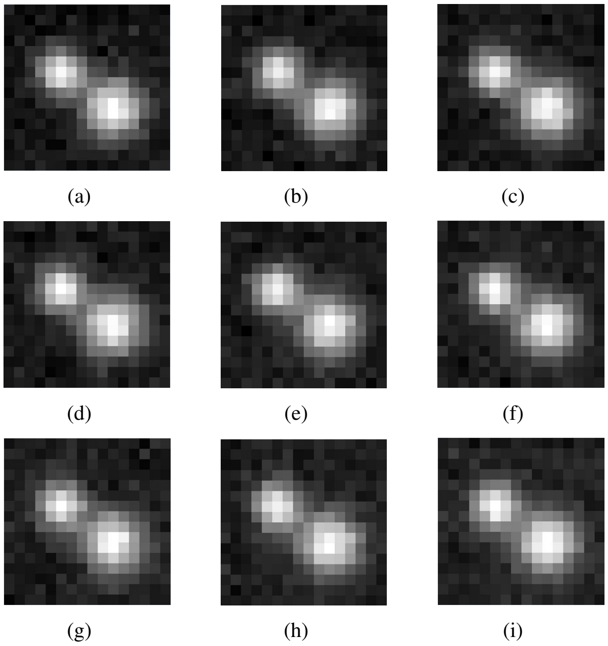


Fig. 9: Centroid estimation error against Gaussian noise for three different algorithms. All these centroid algorithms were evaluated on 20 different star images shown in figure 6(a).


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 10: Examples of samples that we used to evaluate our algorithm.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grants 61231016, 61301193, 61301192 and 61303123.

References

[1] S Thomas, T Fusco, A Tokovinin, M Nicolle, V Michau, and G Rousset, "Comparison of centroid computation algorithms in a shack–hartmann sensor," *Monthly Notices of the Royal Astronomical Society*, vol. 371, no. 1, pp. 323–336, 2006.

[2] Jin-Qiu Sun, Jun Zhou, Zhen Zhang, and Yong-Peng Zhang, "Centroid location for space targets based on

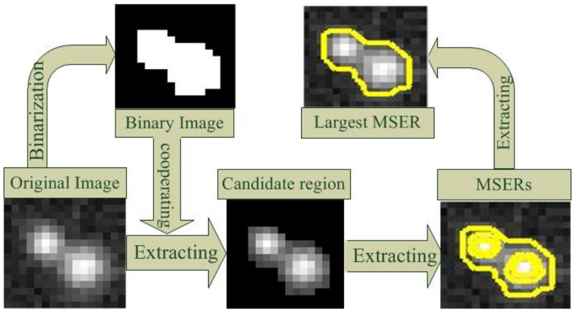


Fig. 11: The process of extracting the largest MSER from star image.

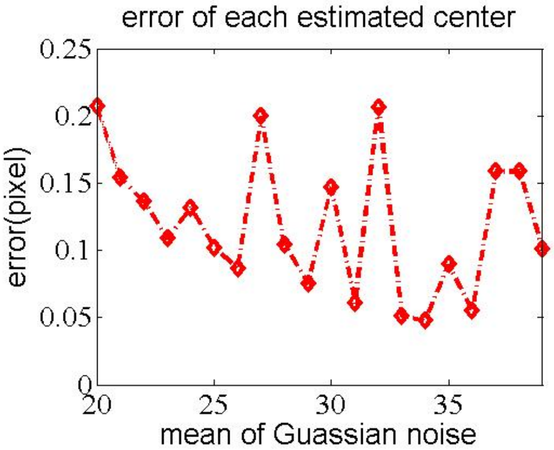


Fig. 12: Centroid estimation error against Gaussian noise for the left star in figure 6(b). The mean and variance of the error is *0.1190* and *0.0510*.

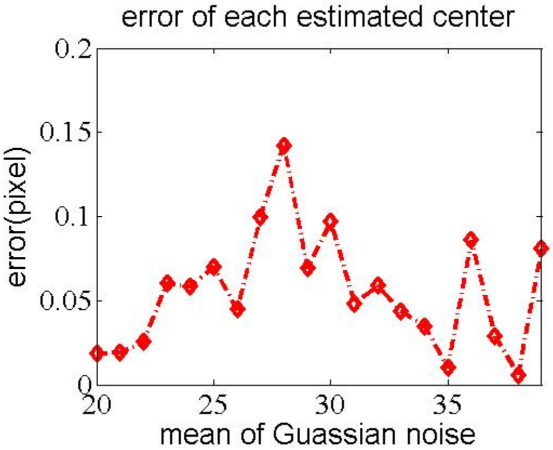


Fig. 13: Centroid estimation error against Gaussian noise for the right star in figure 6(b). The mean and variance of the error is *0.0549* and *0.0346*.

energy accumulation," *Optics and Precision Engineering*, vol. 12, pp. 032, 2011.

[3] Brendan M. Quine, Valery Tarasyuk, Henok Mebrahtu, and Richard Hornsey, "Determining star-image location: A new sub-pixel interpolation technique to process image centroids," *Computer Physics Communications*, vol. 177, no. 9, pp. 700 – 706, 2007.

[4] Huaqiang Li, Helun Song, Changhui Rao, and Xuejun Rao, "Accuracy analysis of centroid calculated by a modified center detection algorithm for shack–hartmann wavefront sensor," *Optics Communications*, vol. 281, no. 4, pp. 750–755, 2008.

[5] Christopher M Bishop et al., *Pattern recognition and machine learning*, vol. 1, springer New York, 2006.

[6] J Matas, O Chum, M Urban, and T Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761 – 767, 2004, British Machine Vision Computing 2002.

[7] Ron Kimmel, Cuiping Zhang, Alexander M Bronstein, and Michael M Bronstein, "Are mser features really interesting?," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 11, pp. 2316–2320, 2011.

[8] Laurent Najman and Michel Couprie, "Building the component tree in quasi-linear time," *Image Processing, IEEE Transactions on*, vol. 15, no. 11, pp. 3531–3539, 2006.

[9] Carlos Arteta, Victor Lempitsky, J Alison Noble, and Andrew Zisserman, "Learning to detect cells using non-overlapping extremal regions," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*, pp. 348–356. Springer, 2012.

[10] Andrea Vedaldi, "An implementation of multi-dimensional maximally stable extremal regions," *Citeseer, Feb*, 2007.

[11] Noam Shental, Aharon Bar-Hillel, Tomer Hertz, and Daphna Weinshall, "Computing gaussian mixture models with em using equivalence constraints.," in *NIPS*, 2003, vol. 50, p. 112.

[12] Richard A Levine and George Casella, "Implementations of the monte carlo em algorithm," *Journal of Computational and Graphical Statistics*, vol. 10, no. 3, pp. 422–439, 2001.

[13] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 881–892, 2002.

[14] Radford M Neal and Geoffrey E Hinton, "A view of the em algorithm that justifies incremental, sparse, and other variants," in *Learning in graphical models*, pp. 355–368. Springer, 1998.