BI-VARIATE STATISTICAL ATTRIBUTE FILTERING: A TOOL FOR ROBUST DETECTION OF FAINT OBJECTS [1]

P. Teeninga[2, 4], U. Moschini[2, 5], S.C. Trager[3, 6], M.H.F. Wilkinson[2, 7]

[2]Johann Bernoulli Institute, University of Groningen, Netherlands,
[3]Kapteyn Astronomical Institute, University of Groningen, Netherlands,
[4]p.d.teeninga@student.rug.nl, [5]u.moschini@rug.nl, [6]s.c.trager@rug.nl,
[7]m.h.f.wilkinson@rug.nl

We present a new method for morphological connected attribute filtering for object detection in astronomical images. In this approach, a threshold is set on one attribute (power), based on its distribution due to noise, as a function of object area. The results show an order of magnitude higher sensitivity than a state-of-the-art method.

Introduction

In astronomy the SExtractor (Source Extractor) [5] is used to detect astronomical sources in images. The problem is threshold selection [6], with the added difficulty that edge information is usually absent. Inspired by similarities between SExtractor and so-called Max-Tree approaches [7], we propose a new scheme for multi-threshold selection and object detection, which varies the threshold depending on the properties of connected components in threshold sets. We introduce a new filtering method, which uses a statistical test on the integrated power within a component, as a function of its area. This sets lower thresholds for extended sources than for compact ones.

SExtractor estimates the local background and noise, and computes a local hierarchy of thresholds, which binarizes objects at a series of levels. Connected components at various threshold levels are considered candidate objects. If a single component at the lowest threshold corresponds to a single component one level higher, both are considered parts of the same object. If multiple components at a higher level are contained within a lower, the object is split by influence zones. The lowest threshold is chosen globally; within each connected component at the lowest threshold level, thresholds are computed depending on the maximum grey level within it. Components below some area threshold are ignored.

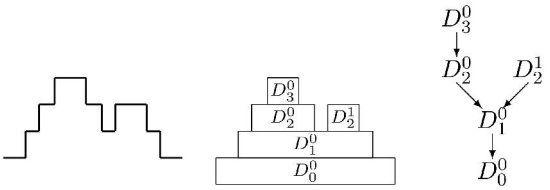


Fig. 1. A 1-D signal $\mathbf{f}$ (left), the corresponding peak components (middle) and the Max-Tree (right). Figure from [8].

The approach used is similar to the Max-Tree approach [7] used in attribute filtering [3]. The difference is that in the Max-Tree all possible thresholds are considered using one of many fast algorithms [2, 7, 8].

A Max-Tree is built from the connected components of all threshold sets of the image. The leaves represent the regional maxima, each node in the tree represents a connected component at a given threshold level, known as a peak component. Since these components are nested, it can easily be seen they form a tree structure. Fig. 1 shows a 1-D example.

During tree construction, one or more attributes can be computed, to allow removal of nodes that fail a criterion. Usually, a single attribute such as area is used, and this is compared to a predefined threshold [3]. In this work we will compute the threshold on one attribute based on the value of another.

In the rest of the paper, we first deal with the problem of background estimation. We then present our object detection method. Finally, we present results and conclusions.

[1] Funded in part by Netherlands Organisation for Scientific Research (NWO) Grant 612.001.110

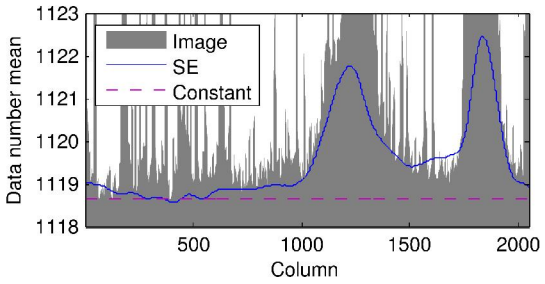


Fig. 2. The bias of SExtractor's background estimation reduces the detectability of low-surface-brightness objects.

Background estimation

The image is the sum of a background *B*, objects image *O*, and Gaussian-like noise where the variance is equal to $g(B+O)+R$, with g the inverse of the gain, and R due to other noise sources. The background has to be estimated and subtracted from the image. As a first approximation, we assume B to be con-stant, which seems to work for the images from the Sloan Digital Sky Survey (SDSS) DR 7 [1]. By contrast, SExtractor uses an adap-tive estimate, which unfortunately correlates strongly with objects present in the image, as shown in Fig. 2. We compute an estimate of B by finding tiles which are devoid of objects, i.e. they are flat. A tile of the image is *flat* if the values are consistent with a single Gaussian distribution. We then approximate constant B by taking the mean value of the flat tiles, which are likely to belong to the background.

The D'Agostino-Pearson K^2 test [4] detects deviations from a Gaussian distribution, but does not take into account the positions of pixels. A slope due to an object might be pre-sent, while the overall distribution ignoring the positions is Gaussian-like. To counter this, t-tests of equal means for different parts of the tile are used as well. We simultaneously estimate the variance in the background regions.

Max-Tree Object Detection

Obviously, a small excursion in brightness over a large area is more significant than a similar excursion in a smaller area. Therefore, we compute the area, simply by counting pixels, and the power [9] given by

$$power(P) = \sum_{x \in P} \left(f(x) - f(parent(P))\right)^2 \qquad (1)$$

for any peak component P, in which f is the grey level of a pixel x. These two combined allow use of a statistical test to determine the significance of a feature found in the Max-Tree.

With the background removed, the variance of the noise is $gO + \sigma_B^2$ where $\sigma_B^2 = gB + R$ is the estimated variance in the background, given our method. The values below 0 are no longer used and are set to 0. In the Max-Tree representation of the image, nodes are marked as significant by **RejectNoise** if it is unlikely that they are due to noise.

Let P be a node with the root as parent and A its area. In this case $f(parent(P)) = 0$. If P is due to noise, the distribution of $f^2(x)/\sigma_B^2$, for x in P is χ^2 with one degree of freedom, and $power(P)/\sigma_B^2$ is χ^2 distributed with A de-grees of freedom which is used in the signifi-cance test. The test is repeated for other nodes by taking the parent node as the local background. The variance in those cases is approximately $g f(parent(P)) + \sigma_B^2$.

The function **RejectNoise** $(M, \alpha, g, \sigma_B^2)$ marks the significance of every node in Max-Tree M, given significance level α, and g and σ_B^2 as before. In the body of the function, we first compute statistic W for every node P, as

$$W = power(P)/\left(g f(parent(P)) + \sigma_B^2\right) \qquad (2)$$

after which a χ^2 significance test is perfor-med. We mark P as significant if rejected.

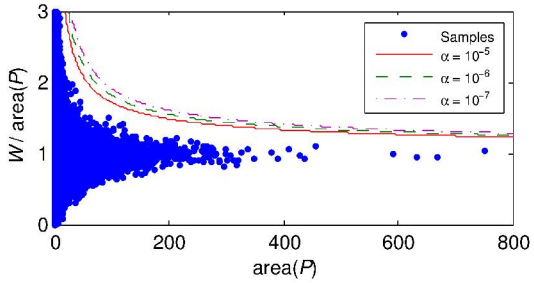


Fig. 3. Rejection boundaries. Includes 50000 node samples taken from a simulated noise image

FindObjects(M) heuristically labels signifi-cant nodes as objects. A significant node P in M represents a part of one or more objects. If P has no significant ancestor at a positive height, then P is labelled as new object. Other-wise, let node Q be the nearest significant ancestor of P along its root path. If there are multiple branches with significant descendants originating at Q, the

branch with the largest area is marked as dominant. If P is not in the dominant branch, it is labelled as a new object.

MoveUp$\left(M,\lambda,g,\sigma_B^2\right)$ is used to move object labels up in the tree by λ times the standard deviation of the noise. The labels are removed when that is not possible. When moving up, the dominant descendant is each time the first choice, as it represents the same object. If no significant descendants exist, then a child with the largest power is chosen.

MTObjects$\left(I,\alpha,\lambda,g,\right)$ combines the support functions to process an image I. A Max-Tree is returned where the nodes corresponding to objects are labelled. In this function we first estimate the background mean and variance. The background is subtracted from the image, and negative values set to zero. We then build a Max-Tree, apply **RejectNoise**, followed by **FindObjects** and **MoveUp**. The resulting tree is returned. If g is not given and the noise is mostly photon noise then it can be estimated by setting R to zero.

Applying a smoothing filter

By default, SExtractor smooths images with a fixed kernel H given by

$$H = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \qquad (3)$$

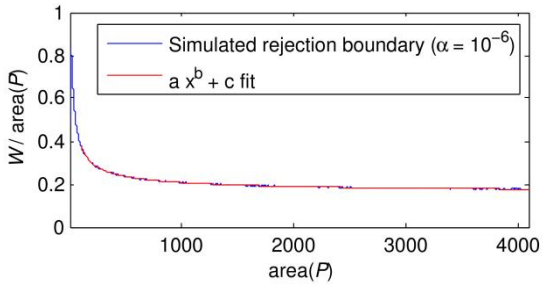


Fig. 4. The rejection boundary computed from simulated noise images for $\alpha = 10^{-6}$.

The χ^2 distribution no longer applies, because of dependency between the pixel values. The rejection boundary can be estimated by filtering unit-variance noise images, and creating Max-Trees for each of them. For every value of area, a histogram of the power is created from the nodes in these trees. The cumulative distribution function (CDF) of the power is estimated from the histogram. The CDF is used to create the rejection boundary shown in Fig. 4. This computation needs to be repeated whenever the filter is changed. The results can be stored for each filter, of course.

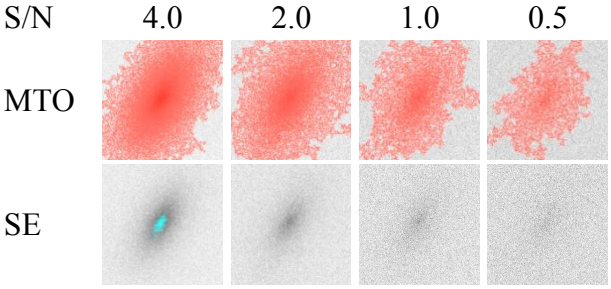


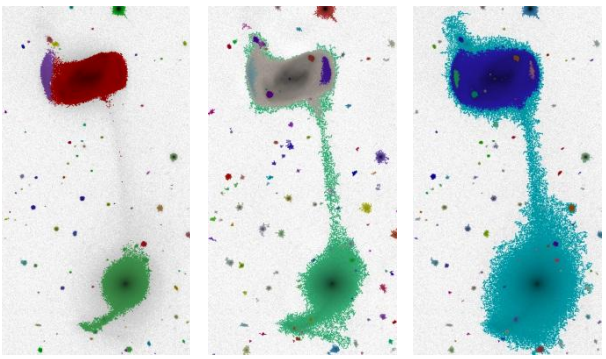
Fig. 5. Results on a 512x512 image of a simulated object with varying S/N. The first row shows results using MTObjects(I, 10^{-6}, 0, -), and the default SExtractor filter. One object is detected in each case. The bottom row shows the results of SExtractor. The object is fragmented at S/N=4.0 and essentially undetected below.

Results

As a first quantitative experiment we generated simulated objects using IRAF, and added noise at different S/N. The results are shown in Fig. 5. Even at S/N=4.0, SExtractor has difficulties in extracting the whole object as a single entity. It detects a few disconnected pixels at 2.0, and fails to detect anything at S/N=1.0. Our method detects the object robustly down to S/N=0.5, so is at least 8 times more sensitive in this case. This difference depends on area, with our method outperforming SExtractor mainly on large objects as those in Fig. 5.

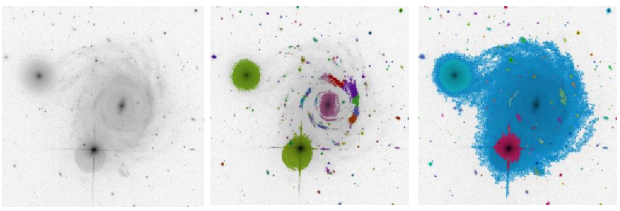
In a more qualitative test on real images, we processed 12 images from the SDSS DR7 [1], each containing a range of objects, including some merging galaxies with faint tidal structures linking them.

An example is shown in Fig. 6. Fig. 6(a) shows the SExtractor result, in which the merging pair is detected as two distinct galaxies. Using the SExtractor background, but combined with MTObjects obtains a much better result in Fig. 6(b), showing them as a single object containing two large components. Combining our background estimate with MTObjects in Fig. 6(c) retains more of the faint outlying regions. This result shows that the improvements seen are due to both the new filtering method and the background estimation.



(a) (b) (c)

Fig. 6. A comparison on a part of fpC-002078-r1-0157.fit: (a) SExtractor; (b) MTObjects(I , 10^{-6}, 0.4, 0.212) using the background image generated by SExtractor; (c) MTObjects(I, 10^{-6}, 0.75, 0.212) using our own background.



(a) (b) (c)

Fig. 7. Result on spiral galaxy: (a) original image; (b) SExtractor result; (c) proposed method.

A full set of 12 images with SExtractor and MTObject results can be inspected on www.cs.rug.nl/~michael/PRIA2013/. Inspection of the results showed no false detections and no objects detected by SExtractor but missed by MTObjects. MTObjects performed much better on extended sources, and in particular face-on spiral galaxies, as seen in Fig 7.

Conclusion

We have presented a new method of object detection using a new approach to attribute filtering, where a threshold is set on one attribute (power), based on its distribution due to noise, as a function of a second attribute (area). The results show an order of magnitude higher sensitivity than the state-of-the-art method implemented in the latest release of SExtractor. The main parameter α controlling the sensitivity of the method is easily interpreted as the probability that a detected object is due to noise. Even at conservative values (10^{-6}) the sensitivity of the current method is excellent. Extended and compound objects are not distorted, as in the SExtractor method. The proposed method is slower than SExtractor, but not prohibitively so, processing a 4 Mpixel image in 1.5 s on an AMD Phenom II X4 955 at 3.2GHz, and scales with $O(N \log N)$, with N the number of pixels [8].

We are working on improvements of the computation of the decision boundary, and parallel implementations, and will extend the experiments to include many more images.

References

1. K.N. Abazajian, J.K. Adelman-McCarthy, M.A. Agüeros, S.S. Allam, C.A. Prieto, D. An, K.S. Anderson, S.F. Anderson, J. Annis, N.A. Bahcall, et al. The seventh data release of the Sloan Digital Sky Survey. // The Astrophysical Journal Supplement Series. - 2009. - Vol. 182. - P. 543-558.
2. C. Berger, T. Geraud, R. Levillain, N. Widynski, A. Baillard, E. Bertin. Effective component tree computation with application to pattern recognition in astronomical imaging // Proc. Int. Conf. Image Proc. 2007. - 2007. Vol. IV, pages 41-44, San Antonio, Texas, USA, September 16-19, 2007.
3. E.J. Breen, R. Jones. Attribute openings, thinnings and granulometries. // Comp. Vis. Image Understand. - 1996. - Vol. 64. - P. 377-389.
4. R.B. D'Agostino, A. Belanger, R.B. D'Agostino Jr. A suggestion for using powerful and informative tests of normality // The American Statistician. - 1990. - Vol. 44. - P. 316-321.
5. E. Bertin, S. Arnouts. Sextractor: software for source extraction // Astronomy & Astrophysics Suppl. - 1996 -Vol. 117. -P. 393-404.
6. P.K. Sahoo, S. Soltani, A.K.C. Wong, Y.C. Chen. A survey of thresholding techniques // Comp. Vision Graph. Image Proc. - 1988. - Vol. 41. - P. 233-260.
7. P. Salembier, A. Oliveras, L. Garrido. Anti-extensive connected operators for image and sequence processing // IEEE Trans. Image Proc. - 1998. - Vol. 7. - P. 555-570.
8. M.H.F. Wilkinson. A fast component-tree algorithm for high dynamic-range images and second generation connectivity // Proc. Int. Conf. Image Proc. 2011. - 2011. - P. 1041-1044.
9. N. Young, A.N. Evans. Psychovisually tuned attribute operators for pre-processing digital video // IEE Proceedings-Vision, Image and Signal Processing,. - 2003.- Vol 150.- P. 277-286.