Improved Detection of Faint Extended Astronomical Objects through Statistical Attribute Filtering

Paul Teeninga[1], Ugo Moschini[1], Scott C. Trager[2], and Michael H.F. Wilkinson[1]

[1]Johann Bernoulli Institute, and [2]Kapteyn Astronomical Institute,
University of Groningen, P.O. Box 407, 9700 AK Groningen, The Netherlands*

Abstract. In astronomy, images are produced by sky surveys containing a large number of objects. SExtractor is a widely used program for automated source extraction and cataloguing but struggles with faint extended sources. Using SExtractor as a reference, the paper describes an improvement of a previous method proposed by the authors. It is a Max-Tree-based method for extraction of faint extended sources without stronger image smoothing. Node filtering depends on the noise distribution of a statistic calculated from attributes. Run times are in the same order.

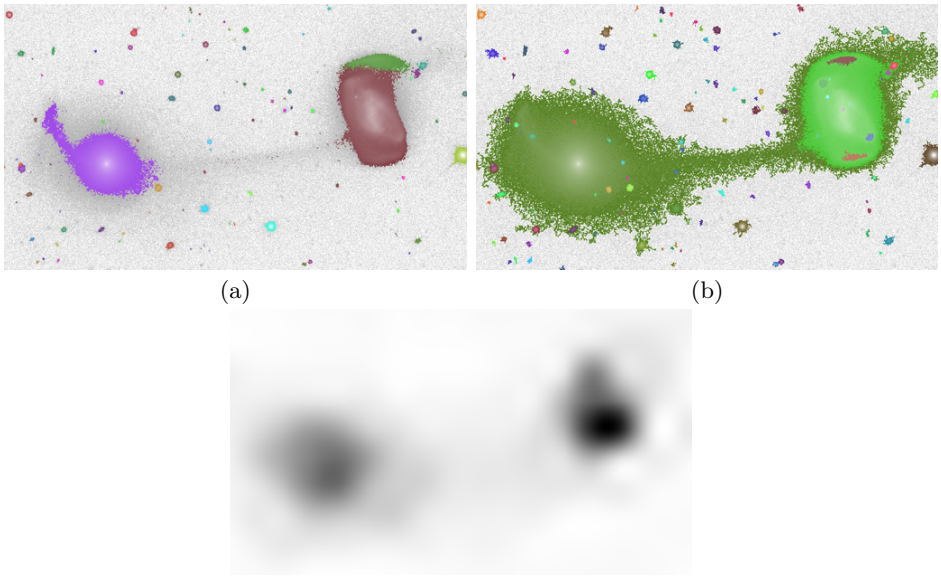
Keywords: Attribute filters, statistical tests, astronomical imaging, object detection

1 Introduction

The processing pipeline of a sky survey includes extraction of objects. With advances in technology more data is available and manually extracting every object is infeasible. A survey example is the Sloan Digital Sky Survey [7] (SDSS) where the DR7 [1] catalogue contains 357 million unique objects. SExtractor [2], a state-of-the-art extraction software, first estimates the image background. With the default settings, to perform a correct segmentation and avoid false positives, objects are identified with the pixels with intensity at a threshold level higher than 1.5 times the standard deviation of the background estimate at that location. We refer here to such mechanism as *fixed threshold*: the threshold value only relies on local background estimates in different sections of the image, ignoring the actual object properties. The downside is that parts of objects below the threshold are discarded (Fig. 1). As improvement to the fixed threshold, we proposed in [8] a method that locally varies the threshold depending on object size by using statistical tests rather than arbitrary thresholds on attributes. In this paper, we present an extension of that method modifying the attribute used. In our method, the supporting data structure is a Max-Tree [5] created

* This work was funded by the Netherlands Organisation for Scientific Research (NWO) under project number 612.001.110.

The final publication is available at link.springer.com.
DOI: 10.1007/978-3-319-18720-4_14.



(a) (b)

(c)

Fig. 1: Result of SExtractor with default settings (a) and the proposed method (b). The filament between the galaxies is not extracted in (a). SExtractor Background estimate (c) shows correlation with objects: a fixed threshold above this already biased estimate would make the segmentation worse.

from the image, where every node corresponds to a connected component for all the threshold levels. The choice is inspired by the simplified component tree used in SExtractor and it was already suggested in [4]. Nodes are marked significant if noise is an unlikely cause, for a given significance level. Nested significant nodes can represent the same object or not and a choice must be made. When deblending objects, other significant branches are considered as new objects. Results are compared with SExtractor. In SExtractor, every connected component above the fixed threshold is re-thresholded at N levels with logarithmic spacing, by default. The range is between the fixed threshold and the maximum value above the component. A tree similar to a Max-Tree is constructed. Branches are considered separate objects if the integrated intensity is above a certain fraction of the total intensity and if another branch with such property exists. An image background, caused by light produced and reflected in earth's atmosphere, is estimated and subtracted before thresholding. SExtractor's estimate shows bias from objects (see Fig. 1), which reduces their intensities. Backgrounds in the SDSS dataset turn out to be nearly flat: constant estimates per image are used. Our background estimate, which also reduces object bias, is briefly described in Section 2. The data set of 254 monochrome images is a subset of the corrected images in SDSS DR7. All selected images contain merging or overlapping galaxies. These often include faint structures which are difficult to detect with

The final publication is available at link.springer.com.
DOI: 10.1007/978-3-319-18720-4_14.

SExtractor. We use r-band images, because they have the best quality [3]. The set has been acquired by CCD. A software bias is subtracted from the images, after which the pixel values are proportional to photo-electron counts [6]. Poissonian noise dominates, but due to the high photon counts at the minimum grey level, the distribution is approximately Gaussian, with a variance which varies linearly with grey level. In the rest of the paper, the extension to our method to identify astronomical objects is described. A comparison with the segmentation performed by SExtractor is presented, followed by conclusions and future directions of research.

2 Background estimation

The image is assumed to be the sum of a background image B, object image O and Gaussian noise. The noise variance is equal to $g^{-1}(B+O)+R$, with g equivalent to `gain` in the SDSS catalogue, and R due to other noise sources, such as read noise, dark current and quantisation. First, the image background must be computed. A method giving a constant estimate for the background value that does not correlate with the objects was proposed in [8] and explained more extensively in [9]. With the background removed, the variance of the noise is $g^{-1}O+\sigma_{\mathrm{bg}}^2$, where $\sigma_{\mathrm{bg}}^2 = g^{-1}B+R$ is approximately equal to our estimate. With $\hat{\mu}_{\mathrm{bg}}$ and $\hat{\sigma}_{\mathrm{bg}}^2$ the estimates of background and variance, respectively, in case g is not given, it is approximated by $\hat{\mu}_{\mathrm{bg}}/\hat{\sigma}_{\mathrm{bg}}^2$. It is assumed that R is small compared to $g^{-1}\hat{\mu}_{\mathrm{bg}}$. Negative image values are set to 0 and the Max-Tree is constructed. The next step is to identify nodes that are part of objects, referred to as *significant* nodes.

3 Identifying significant nodes

To identify the nodes in the tree belonging to objects, let us define P_{anc} as the closest significant ancestor or, if no such node exists, the root, for a node P in the Max-Tree. P is considered significant if it can be shown that $O(x) > f(P_{\mathrm{anc}})$ for pixels $x \in P$, given a significance level α. The pixel value associated with P is indicated with $f(P)$. Similarly, let $f(x)$ be the value of pixel x. The power [11] attribute is similar to the definition of object *flux* or the integrated intensity that also SExtractor uses. It is a measure widely used for object identification in astronomy. A definition of the power attribute of P is

$$\mathrm{power}(P) := \sum_{x \in P} (f(x) - f(\mathrm{parent}(P)))^2. \tag{1}$$

An alternative definition that will be used is

$$\mathrm{powerAlt}(P) := \sum_{x \in P} (f(x) - f(P_{\mathrm{anc}}))^2. \tag{2}$$

To normalize the power attribute, the values are divided by the noise variance σ^2. Four significance tests that use the attributes above are defined in the following.

The final publication is available at link.springer.com.
DOI: 10.1007/978-3-319-18720-4_14.

Significance test 1: `power` *given* `area` *of the node.* For node P we use hypothesis

$$H_{\texttt{power}} := (O(x) \leq f(\text{parent}(P)) \quad \forall x \in P.$$

This test uses the definition of power attribute in Equation 1. Let us assume $H_{\texttt{power}}$ is true and consider the extreme case $O(x) = f(\text{parent}(P))$ for pixels $x \in \text{parent}(P)$. The variance of the noise σ^2 is equal to $g^{-1}f(\text{parent}(P)) + \sigma^2_{\text{bg}}$. For a random pixel x in P, the value $(f(x) - f(\text{parent}(P)))^2/\sigma^2$ has a χ^2 distribution with 1 degree of freedom. As the pixel values in P are independent, $\texttt{power}(P)/\sigma^2$ has a χ^2 distribution with degrees of freedom equals to the area of P. The χ^2 CDF (or inverse), available in scientific libraries, can be used to test the normalized `power` given significance level α and node area. An example of a rejection boundary of a χ^2 CDF is shown in Fig. 2a. If $\texttt{power}(P)/\sigma^2 > \texttt{inverse}\chi^2\texttt{CDF}(\alpha, \texttt{area})$, $H_{\texttt{power}}$ is rejected: $O(x) > f(\text{parent}(P)) \geq f(P_{\text{anc}})$, for pixels $x \in \text{parent}(P)$, making P significant.

In all the next three significance tests (all right tailed) the exact distribution of `powerAlt` is not known and it is obtained by simulation. In general, the rejection boundary for a generic `attribute` and significance level α is the result of the inverse CDF, which will be denoted as $\texttt{inverseAttributeCDF}(\alpha, ...)$. Let r be an integer greater than zero and $n_{\text{samples}}(\alpha, r) := r/\alpha$ with n_{samples} rounded to the closest integer. A number of random independent nodes equals to n_{samples} is generated. On average, for r nodes the `attribute` value is greater than or equal to the rejection boundary. The best estimate of $\texttt{inverseAttributeCDF}(\alpha, ...)$, without any further information about the distribution, is the average of the two smallest of the $r + 1$ largest `attribute` values.

Significance test 2: `powerAlt` *given* `area` *and* `distance`. For node P we now use hypothesis

$$H_{\texttt{powerAlt}} := (O(x) \leq f(P_{\text{anc}})) \quad \forall x \in P.$$

In significance test 1, leaf nodes are less likely to be found significant due to their small area and the low intensity difference with the parent node. The main idea behind significance test 2 is to make the significance level more constant for every node, independently of its height in the tree, by referring to its ancestor rather than the parent node. The definition of power attribute in Equation 2 is used. Let us assume $H_{\texttt{powerAlt}}$ is true and consider the extreme case $O(x) = f(P_{\text{anc}})$. Let us define $\texttt{distance}(P) := f(P) - f(P_{\text{anc}})$. Let X be a random set of $\texttt{area}(\text{P}) - 1$ values drawn from a truncated normal distribution with a minimum value of $\texttt{distance}(P)$. The variance of the normal distribution is set to $\sigma^2 = g^{-1}f(P_{\text{anc}}) + \sigma^2_{\text{bg}}$. Attribute $\texttt{powerAlt}(P)$ has the same distribution as $\texttt{distance}^2(P)$ plus the sum of the squared values in X. The rejection boundary for the normalized `powerAlt` attribute is provided by the function $\texttt{inversePower-AltCDF}(\alpha, \texttt{area}, \texttt{distance})$. Hypothesis $H_{\texttt{powerAlt}}$ is rejected if $\texttt{powerAlt}(P)/\sigma^2 > \texttt{inversePowerAltCDF}(\alpha, \texttt{area}, d))$: $O(x)$ at some pixels x in P is higher than $f(P_{\text{anc}})$, making P significant. The minimum area of a significant node is 2 pixels. An estimate is given for `inversePowerAltCDF` for

The final publication is available at link.springer.com.
DOI: 10.1007/978-3-319-18720-4_14.

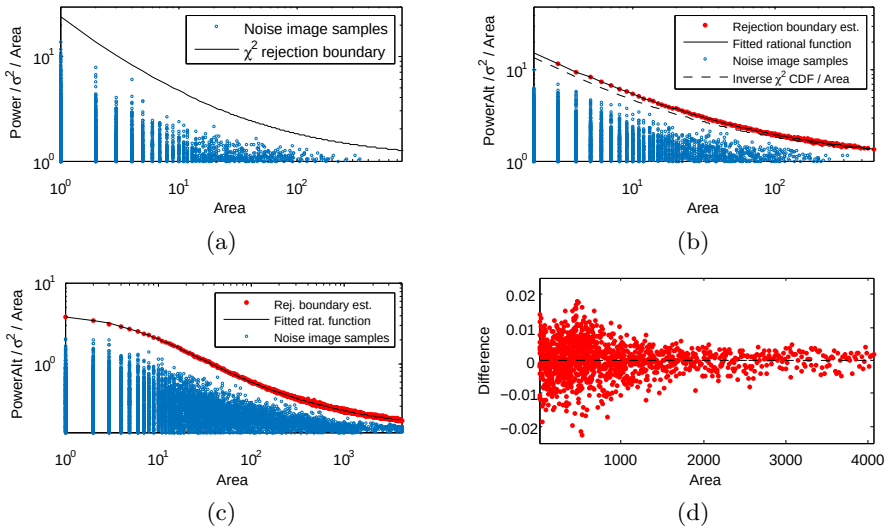


Fig. 2: Rejection boundaries for significance test 1 (a) and the simulated rejection boundaries for test 3 (b) and test 4 (c). Log-log scale. (d) shows the difference between the rational function and the estimate in (c). $\alpha = 10^{-6}$.

constant α, varying `area` and `distance`. For each rejection boundary, varying `distance`, a rational function is fitted to reduce the error and the storage space.

Significance test 3: `powerAlt` *given* `area`. It is a significance test that has the same goal of significant test 2. It uses the distribution of `powerAlt` given α, `area`. It is independent of the `distance` measure, not used as parameter in the inverse CDF. Using the assumptions from *significance test 2*, `distance`(P) has a truncated normal distribution with a minimum value of 0, the same distribution as a random non-negative pixel value. The rejection boundary is calculated through simulated images. Fig. 2b shows the rejection boundary and the fitted rational function for this significance test.

Significance test 4: `powerAlt` *given* `area`*, using a smoothing filter.* It is basically equal to significance test 3 with the addition of the smoothing. Smoothing reduces noise. A larger number of objects is detected with this test. The default smoothing filter used in SExtractor is used as in [10]. Filtering is done after background subtraction and before setting negative values to zero. Fig. 2c shows the rejection boundary with its fitted rational function and Fig. 2d shows the difference between the rational function approximation and the estimates for this significance test.

Considerations about the false positives. Alg. 1 describes the method used for marking nodes as significant. Visiting nodes in non-decreasing order by pixel

The final publication is available at link.springer.com.
DOI: 10.1007/978-3-319-18720-4_14.

Algorithm 1 `SignificantNodes`$(M, \texttt{nodeTest}, \alpha, g, \sigma^2_{\text{bg}})$

In: Max-Tree M, significance test t, significance level α, gain g, variance of the noise at the background σ^2_B.
Out: Nodes in M that are unlikely to be noise are marked as significant.
1: **for all** nodes P in M with $f(P) > 0$ in non-decreasing order **do**
2: **if** `nodeTest`$(M, P, \alpha, g, \sigma^2_{\text{bg}})$ is `true` **then**
3: Mark P as significant.
4: **end if**
5: **end for**

value simplifies the identification of $P_{\texttt{anc}}$, if stored for every node. There is no need if the χ^2 test is used. Function `nodeTest`$(M, P, \alpha, g, \sigma^2_{\text{bg}})$ performs the significance test and returns true if P is significant, false otherwise. The Max-Tree of a noise image after subtraction of the mean and truncation of negative values is expected to have $0.5n$ nodes, with n the number of pixels. An upper bound on the number of expected false positives is $\alpha 0.5n$ if the nodes are independent, which is not entirely the case. Given a 1489×2048 noise image, the same size of the images in the data set, and $\alpha = 10^{-6}$, the upper bound on the expected number of false positives is approximately 1.52. An estimate of the actual number of false positives is 0.41, 0.72, 0.94 and 0.35 for the four significance tests, respectively, averaged over 1000 simulated noise images.

4 Finding the objects

Multiple significant nodes could be part of the same object. A significant node with no significant ancestor is marked as an object. Let `mainBranch`(P) be a significant descendant of P with the largest area. A significant node, with significant ancestor P_{anc}, that differs from `mainBranch`(P_{anc}) is marked as a new object. The assumption of a new object is not valid in every case: it depends on the used significance test, filter and connectivity, as it will be shown in the comparison section. The method is described in Alg. 2.

4.1 Moving up object markers

Nodes marked as objects have a number of pixels attached due to noise. The number decreases at a further distance from the noiseless image signal, which can be achieved by moving the object marker up in the tree, λ times the standard deviation of the noise. The obvious choice for an object node P is `mainBranch`(P), if such a node exists, since it does not conflict with other object markers. Otherwise, it is chosen the descendant of P with the highest p-value found with the related CDF for the `power` or `powerAlt` attribute. However, the CDF is not always available or easy to store. Instead, the descendant with the largest `power` attribute is chosen, if at least one exists. The function that returns the descendant is called `mainPowerBranch`(P). Alg. 3 describes the method. An alternative to allowing a lower value of $f(P_{\text{final}})$ is to remove those object markers.

The final publication is available at link.springer.com.
DOI: 10.1007/978-3-319-18720-4_14.

Algorithm 2 `FindObjects`(M)

In: Max-Tree M.
Out: Nodes in M that represent an object are marked.

for all significant nodes P in M **do**
  if P has no significant ancestor **then**
    Mark P as object.
  else
    if `mainBranch`(P_{anc}) does not equal P **then**
      Mark P as object.
    end if
  end if
end for

Algorithm 3 `MoveUp`$(M, \lambda, g, \sigma^2_{\text{bg}})$

In: Max-Tree M, factor λ, gain g, variance of the noise at the background σ^2_{bg}.
Out: For every object marker that starts in a node P and moves to P_{final}: $f(P_{\text{final}}) \geq f(P_{\text{anc}}) + \lambda$ times the local standard deviation of the noise, when possible. $f(P_{\text{final}})$ might be lower if P_{final} has no descendants.

for all nodes P in M marked as objects **do**
  Remove the object marker from P.
  $h \leftarrow f(P_{\text{anc}}) + \lambda\sqrt{g^{-1}f(P_{\text{anc}}) + \sigma^2_{\text{bg}}}$
  while $f(P) < h$ **do**
    if P has a significant descendant **then**
      $P \leftarrow$ `mainBranch`(P).
    else if P has a descendant **then**
      $P \leftarrow$ `mainPowerBranch`(P).
    else
      Break.
    end if
  end while
  Mark P as object.
end for

If the parameter λ is set too low, there are too many noise pixels attached to objects. However, to be able to display faint parts of extended sources a low λ is preferred. After thorough tests on objects simulated with the IRAF software with several noise levels, a value of 0.5 was found to be a good compromise. Alg. 4 summarises the whole procedure from background estimation to object identification. The proposed method is called MTObjects.

5 Comparison with Source Extractor

Our method is compared against the segmentation performed by SExtractor 2.19.5. The settings are kept close to their default values:

The final publication is available at link.springer.com.
DOI: 10.1007/978-3-319-18720-4_14.

Algorithm 4 `MTObjects`$(I, \texttt{nodeTest}, \alpha, g, \lambda)$

In: Image I, function `nodeTest`, significance level α, gain g and move factor λ.
Out: Max-Tree M. Nodes in M corresponding to objects are marked.

1: $(\hat{\mu}_{\text{bg}}, \hat{\sigma}^2_{\text{bg}}) \leftarrow$ `EstimateBackgroundMeanValueAndVariance`()
2: $I_{i,j} \leftarrow \max(I_{i,j} - \hat{\mu}_{\text{bg}}, 0)$
3: $M \leftarrow$ create a Max-Tree representation of I.
4: `SignificantNodes`$(M, \texttt{nodeTest}, \alpha_{\text{nodes}}, g, \hat{\sigma}^2_{\text{bg}})$.
5: `FindObjects`(M).
6: `MoveUp`$(M, \lambda, g, \hat{\sigma}^2_{\text{bg}})$.

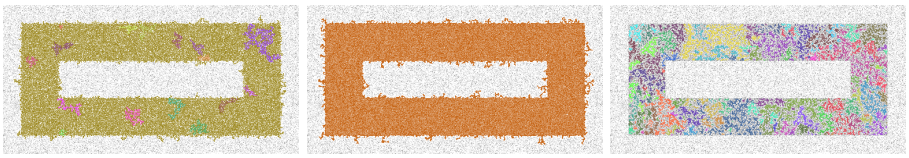


Fig. 3: Fragmented simulated object. Significance test 3 (left), test 4 (middle) and SExtractor (right). The pixels of the object have the value 1.5, close to the SExtractor threshold. Background is 0. Gaussian noise is added with $\sigma = 1$.

- Our background and noise root mean square estimates are used. This already improves the segmentation of SExtractor with respect to the original estimate of SExtractor, that correlates too much with objects.
- `DETECT_MINAREA = 3`. Default in SExtractor 2.19.5.
- `FILTER_NAME = default.conv`.
- `DETECT_THRESH = 1.575`σ above the local background. The default threshold of 1.5 is changed to make the expected false positives similar to significance test 4 for noise-only images. Expected false positives per image is approximately 0.38 based on the results of 1000 simulated noise images.

While there is no guarantee that these settings are optimal, our comparison gives an impression of the performance of our method. A more comprehensive comparison test is required.

Object fragmentation Another source of false positives is fragmentation of objects due to noise. An example is shown in Fig. 3. Fragmentation appears to happen in relatively flat structures and the chance is increased if different parts of the structure are thinly connected. If one pixel connects two parts, the variation in value due to noise can make a deep cut. In the case of the threshold used by SExtractor, fragmentation is severe if the object values are just below the threshold. The expected number of false positives due to fragmentation for the given data set is unknown. Most images do not clearly show fragmented objects. An image where it does happen is displayed in Fig. 4. While the SExtractor parameter `CLEAN_PARAM` can be changed to prevent this from happening, it is left to the default as it has a negative effect on the number of objects detected and it causes fragmentation in this image only.

The final publication is available at link.springer.com.
DOI: 10.1007/978-3-319-18720-4_14.

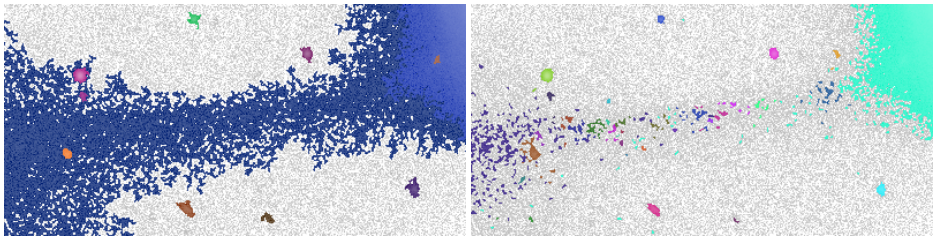
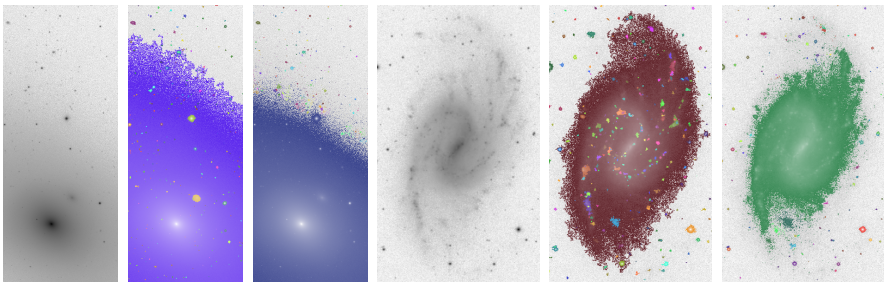


Fig. 4: Fragmentation of an object. Significance test 4 (left) and SExtractor (right). Crop section of `fpC-002078-r1-0157.fits`.



(a) (b) (c) (d) (e) (f)

Fig. 5: MTObjects finds more nested objects. Crop of `fpC-003804-r5-0192.fits`: (a) original image; (b) significance test 4; (c) SExtractor. Crop of `fpC-001332-r4-0066.fits`: (d) original image; (e) significance test 4; (f) SExtractor.

Object count All the significance tests were compared against each other and SExtractor. The significance test 4 returns a larger number of objects in about 100% of the images in the dataset w.r.t. significance test 1 and 2 and in about 70% w.r.t. significance test 3 and SExtractor. After inspection of the results, we noticed that MTObjects detects more objects nested in larger objects (galaxies), when the pixel values of the nested objects are above the SExtractor's threshold. Examples are shown in Fig. 5. This is explained by the fact that every node in the Max-Tree is used, while SExtractor uses a fixed number of sub-thresholds. A question is if the better detection of nested objects can explain the performance of significance test 4 compared to SExtractor. To answer that, the data set is limited to smaller objects. This is done by making a sorted area list of the largest connected component of each image at the threshold used by SExtractor. The performance of significance test 4 and SExtractor is now indeed similar: it means that the difference in the total number of all the objects found in the images is explained by the number of nested object detections. We tested then how significance test 4 performs in the case of densely spaced overlapping objects. When two identical objects overlap, one of the nodes marked as object has a lower `power` or `powerAlt` value on average. If overlapping objects are close enough to

The final publication is available at link.springer.com.
DOI: 10.1007/978-3-319-18720-4_14.

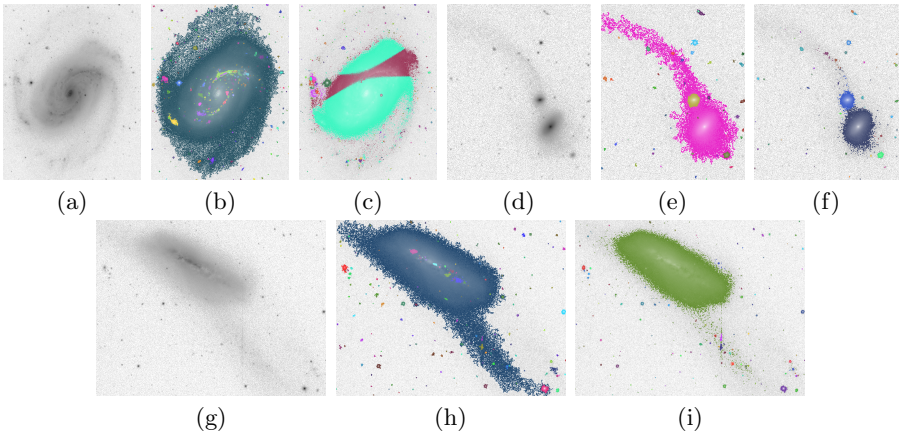


Fig. 6: Comparison of objects with faint extended regions; (a,b,c) crop of `fpC-003903-r2-0154.fits`: (a) original image; (b) significance test 4; (c) SExtractor; (d,e,f) crop of `fpC-004576-r2-0245.fits`: (d) original image; (e) significance test 4; (f) SExtractor; (g,h,i) crop of `fpC-004623-r4-0202.fits`: (g) original image; (h) significance test 4; (i) SExtractor.

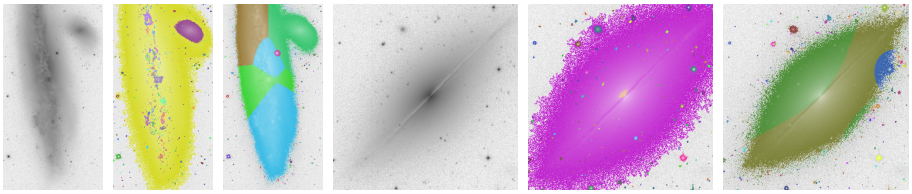
each other and at SExtractor's threshold they are still detected as separate objects, MTObjects could fail to detect them as separate objects. A grid filled with small stars is generated with the IRAF software. The magnitude is set to -0.2 to make objects barely detectable when noise is added. The diameter of objects is 3 pixels (full width at half maximum). The background equals 1000 at every pixel and the gain is 1. Gaussian noise is added. In this case of densely spaced objects, SExtractors detects a number of stars closer to the actual number than MTObjects with significance test 4.

Faint structures The parameter λ used in `MoveUp` is set to 0.5 without adding false positives, which makes detecting fainter structures easier. It is possible to lower it further at the price of more noise included in the segmentation. Object deblending by SExtractor does not always work well if objects do not have a Gaussian profile (Fig. 6).

Dust lanes and artifacts Dust lanes as in Fig. 7 and artefacts as in Fig. 8 are also a source of false positives. In Fig. 7(f), the galactic core is clearly split due to dust. Fig. 8(a)(b)(c) could represent an artefact or a vertical cut-off. Refraction spikes shown in Fig. 8(d)(e)(f) can also cause false positives as in the wave-like shape in Fig 8(f).

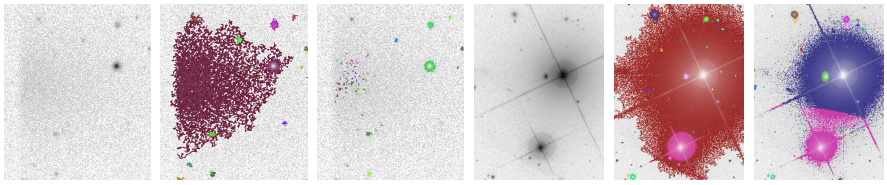
Run times The timer is started before background estimation and is stopped after object classification in SExtractor and after executing `MoveUp` in MTObjects. The amount of time spent on classification in SExtractor is unknown. MTObjects does not perform any classification. Tests were done on an AMD

The final publication is available at link.springer.com.
DOI: 10.1007/978-3-319-18720-4_14.



(a) (b) (c) (d) (e) (f)

Fig. 7: Dust lanes. Crop of `fpC-004623-r4-0202.fits`: (a) original image; (b) significance test 4; (c) SExtractor. Crop of `fpC-001739-r60308.fits`: (d) original image; (e) significance test 4; (f) SExtractor.



(a) (b) (c) (d) (e) (f)

Fig. 8: Artefacts. Crop of `fpC-002326-r4-0174.fits`:(a) original image; (b) significance test 4; (c) SExtractor. Crop of `fpC-001345-r3-0182.fits`: (d) original image; (e) significance test 4; (f) SExtractor.

Phenom II X4 955. SExtractor is typically 3.6 times faster than MTObjects, but it takes longer time for images that have many pixel values above the fixed threshold. MTObjects is more constant in run time.

6 Conclusions and future work

The Max-Tree method (MTObjects) performs better at extracting faint parts of objects compared to state-of-the-art methods like SExtractor. The sensitivity increases with object size. MTObjects appears to be slightly worse in case of densely spaced and overlapping objects, like globular clusters. When an object is defined to have a single maximum pixel value, excluding maxima due to noise, MTObjects is better at finding nested objects. Every possible threshold is tested in MTObjects, whereas SExtractor is bound to a fixed number of thresholds. Deblending objects appears to be better in MTObjects when there is a large difference in size and objects do not have a Gaussian profile. Otherwise, one of the objects will be considered as a smaller branch by MTObjects. Too many pixels are assigned arbitrarily to a single object. The SExtractor method of fitting Gaussian profiles makes more sense in this case and allows for a more even split in pixels. This method could be added as postprocessing step to MTObjects.

The power attribute was initially chosen because in the non-filtered case it has a known scaled χ^2 distribution. Better attribute choices could be investigated.

The final publication is available at link.springer.com.
DOI: 10.1007/978-3-319-18720-4_14.

Deblending similar sized objects can be improved. Nested significant connected components can represent the same object. The current choice, controlled by λ in `MoveUp` is not ideal. The threshold looks too high for large objects and too low for small objects. Parameter λ could be made variable and dependant on the filter, connectivity and node attributes used. If other noise models are used in other data sets, significance tests should be adjusted accordingly. The degree of smoothing applied that helps to avoid fragmentation could be further investigated. Currently, the rejection boundaries are approximated by simulations which must be recomputed for every filter and significance level. Knowing the exact distributions will speed up this phase.

References

1. Abazajian, K.N., Adelman-McCarthy, J.K., Agüeros, M.A., Allam, S.S., Prieto, C.A., An, D., Anderson, K.S.J., Anderson, S.F., Annis, J., Bahcall, N.A., et al.: The seventh data release of the sloan digital sky survey. The Astrophysical Journal Supplement Series 182(2), 543 (2009)
2. Bertin, E., Arnouts, S.: Sextractor: Software for source extraction. Astronomy and Astrophysics Supplement Series 117, 393–404 (1996)
3. Gunn, J.E., Carr, M., Rockosi, C., Sekiguchi, M., Berry, K., Elms, B., De Haas, E., Ivezić, Ž., Knapp, G., Lupton, R., et al.: The Sloan digital sky survey photometric camera. The Astronomical Journal 116(6), 3040 (1998)
4. Perret, B., Lefevre, S., Collet, C., Slezak, E.: Connected component trees for multivariate image processing and applications in astronomy. In: Pattern Recognition (ICPR), 2010 20th International Conference on. pp. 4089–4092 (Aug 2010)
5. Salembier, P., Oliveras, A., Garrido, L.: Antiextensive connected operators for image and sequence processing. Image Processing, IEEE Transactions on 7(4), 555–570 (1998)
6. sdss.org: Photometric flux calibration, `http://www.sdss2.org/dr7/algorithms/fluxcal.html`, published online: http://www.sdss2.org/dr7/algorithms/fluxcal.html
7. Stoughton, C., Lupton, R.H., Bernardi, M., Blanton, M.R., Burles, S., Castander, F.J., Connolly, A.J., Eisenstein, D.J., Frieman, J.A., Hennessy, G.S., et al.: Sloan digital sky survey: early data release. The Astronomical Journal 123(1), 485 (2002)
8. Teeninga, P., Moschini, U., Trager, S.C., Wilkinson, M.H.F.: Bi-variate statistical attribute filtering: A tool for robust detection of faint objects. In: 11th International Conference "Pattern Recognition and Image Analysis: New Information Technologies" (PRIA-11-2013). pp. 746–749 (2013)
9. Teeninga, P., Moschini, U., Trager, S.C., Wilkinson, M.H.F.: Improving background estimation for faint astronomical object detection. In: Proc. ICIP 2015 (2015), submitted.
10. Teeninga, P., Moschini, U., Trager, S.C., Wilkinson, M.H.F.: Bi-variate statistical attribute filtering: a tool for robust detection of faint objects. In: 11th International Conference "Pattern Recognition and Image Analysis: New Information Technologies" (PRIA-11-2013). pp. 746–749. IPSI RAS (2013)
11. Young, N., Evans, A.N.: Psychovisually tuned attribute operators for preprocessing digital video. IEE Proceedings-Vision, Image and Signal Processing 150(5), 277–286 (2003)

The final publication is available at link.springer.com.
DOI: 10.1007/978-3-319-18720-4_14.