Universidad de La Laguna

Trabajo de Fin de Grado

Finding Low Surface Brightness Galaxies near NGC1042 using GNU Astronomy Utilities

Búsqueda de galaxias de bajo brillo superficial cercanas a NGC1042 usando GNU Astronomy Utilities

Andrés García-Serra Romero

Tutorizado por:

Dr. Ignacio Trujillo

Grado en Física

2020-2021

ABSTRACT

This document describes the detection of Low Surface Brightness Galaxy (LSBG) satellites around NGC1042 using the data obtained by the *LBT Imaging of Galaxy Halos and Tidal Structures* (LIGHTS) survey. This survey has been recently proposed by a team of IAC researchers in collaboration with other institutes with the objective of studying the low surface brightness universe for a better understanding on the behavior of galaxy stellar halos and the commonly known "missing satellites problem". This last topic is the principal subject of this work. In collaboration with some members of the LIGHTS team, the project consisted of the development of an algorithm capable of detecting these very low surface brightness objects. The document will start introducing the project and the data set obtained, then the process of detection and categorization of the galaxies will be explained, which will let to a final sample of these objects. Throughout the document some discussions will be made about the difficulties and challenges behind the observation and detections of these very faint structures. For this analysis some objects detected previously in the literature have been used as a reference.

RESUMEN

En este documento se presentan algunas candidatas a Galaxias de Bajo Brillo Superficial (LSBG) alrededor de NGC1042 usando datos del cartografiado *LBT Imaging of Galaxy Halos and Tidal Structures* (LIGHTS). Este cartografiado ha sido propuesto recientemente por investigadores del IAC en colaboración con otras instituciones con el objetivo principal de estudiar el universo de bajo brillo superficial para entender más a fondo el comportamiento de halos estelares de galaxias y el "problema de los satélites perdidos"; siendo este último el principal tema por el que se desasrrolla este trabajo. En colaboración con algunos miembros del equipo de LIGHTS, el trabajo ha consistido en desarrollar un algoritmo capaz de detectar estos objetos de muy bajo brillo superficial. Se comenzará el documento introduciendo el proyecto y los datos obtenidos, después se desarrollará el proceso de detección y categorización de estas galaxias, lo que llevará finalmente a un muestreo de estas. Durante el proceso se discutirán las dificultades y retos detrás de la observación y detección de estos objetos usando en todo momento como referencia detecciones anteriores presentes en la bibliografía.

Contents

Chapter 1

Introduction

En este primer capítulo se introduce la situación actual de la astronomía de bajo brillo superficial como consecuencia del desarrollo tecnológico reciente en instrumentación astronómica. Se detalla la complejidad de observar a brillos superficiales tan débiles y se nombran algunos de los campos dentro de la astrofísica que han podido aprovechar estas mejoras tecnológicas. Posteriormente se explica el "problema de los satélites perdidos", donde se exponen las diferentes soluciones teóricas que se han planteado al problema, así como algunos estudios recientes que se han desarollado en torno a este. Esto se enlaza con la temática del propio TFG, que busca crear una herramienta que permita la detección de estos satélites alrededor de una galaxia de interés. Finalmente se explica la metodología seguida en el trabajo y se exponen unos objetivos iniciales como forma de poder evaluar el progreso a lo largo de estos meses.

As astronomical instrumentation continues its development, a new era is cutting through in astronomical research scoping for new imaging never achieved before. For the first time in history, astronomers are capable of detecting objects that have a surface brightness 5000 times fainter than the darkest night sky on Earth ($\sim$ 22 mag/arcsec $^{-2}$).

All this has opened research in different matters, such as the effect of dark matter on the formation of the most massive and also on the smallest structures in the Universe. Some of these studies have found, with this deep imaging, a large population of Ultra Diffuse Galaxies, whose nature is still being studied.

Related to this, one of the scientific challenges that has re-emerged is the so-called "missing satellites problem". This is the discrepancy found when comparing the ΛCDM cosmological simulations on galaxy formation that result in a much higher number of galaxy subhalos than the number of satellites we are able to observe.

Several ideas have been suggested to solve the above discrepancy. For example, if these small galaxies are not massive enough to retain barionic matter, such as stars, or gas that is needed to form stars, then these galaxies would not be easily visible. Another explanation is that these dwarf galaxies are so small that they merge into the galaxies that they orbit shortly after they are formed. Lastly, it could also be that the dwarf galaxies are so small that, even if they are not merged by the galaxies they orbit, tidal interactions due to the gravitational

pull of the galaxy could strip away the stars leaving diffuse low surface brightness galaxies that are extremely difficult to find. Some astronomers also believe that the star formation mechanisms behind these dwarf galaxies are the key of understanding the reason why they are not being seen.

In the last years, a lot of research has been made around this. Some studies have focused on our own Galaxy (see McConnachie (2012)) while others have examined around other members of the Local Group like M31 (see Martin et al. (2016), McConnachie et al. (2018)).

But to have a larger perspective of how this cosmological problem behaves it is desirable to approach the discrepancy with a wider study, that is why it is necessary to explore the population of the satellites around galaxies beyond the Local Group and with different morphologies.

Recently, a group of IAC researchers has shown that with ultra-deep imaging from large current telescopes it is possible to detect these objects with more accuracy than ever before. Observing at such deep magnitudes, the contrast of low surface brightness objects should increase relative to the background galaxy population, giving an opportunity of studying this "missing satellites problem" at long distances, far away from the Local Group. This will be presented shortly in an upcoming paper (submitted, Trujillo et al. (2021)), in which the *LBT Imaging of Galactic Halos and Tidal Structures* (LIGHTS) survey will be introduced.

1.1 My project

All of this leads to this end-of-degree project. This project will focus on making an algorithm that automatizes the detection of these low surface brightness satellites using GNU Astronomy Utilities[1] (`gnuastro` from now on).[2]

To make this project possible, I firstly had to learn from zero how to use the Linux environment and all the `gnuastro` tools I needed flawlessly. That is why my first months of research and work were focused on learning how to code in shell and getting used to working with `gnuastro`. In these months I started making detections on HST[3] public data and made some catalogues of interest to get used to these new tools.

The next chapters will describe, analyze and discuss all the steps that made possible the final algorithm. We will first introduce the data acquisition, the instrumentation needed for such faint features and then jump right into the detection, classification and measurements of the objects of interest, giving a sample of these objects detected by the algorithm at the end of the document.

[1] `gnuastro` is registered on the Astrophysics Source Code Library (ASCL) with the identifier 1801.009

[2] All the code scripts and pipeline used in this work can be publicly accessed in: `https://gitlab.com/AndyGarciaSerra/trabajo-de-fin-de-grado`

[3] Acronym for *Hubble Space Telescope*, this 2.4m telescope is an international project sent to space at 1990. It orbits Earth at approximately 500km above the ground to avoid atmosphere contamination.

1.2 Objectives

In every scientific journey it is important to set some goals to have a way to measure your achievements and progress. In this short section we will discuss some.

First of all, in the academic field the goals are clear, learn how to work in the Linux environment and get used to the `gnuastro` tools and shell scripting. Then, elaborate a script that is capable of detecting, given a reduced image, the maximum number of these objects of interest possible, separating them of other kind of similar objects and cataloguing them, being able to give a sample of these objects and a table of measurements for each one.

In the personal field, I would like to learn how to work as a team member collaborating with the LSB group of the IAC and meeting each week with them sharing my progress and hopefully helping other students. Also, learning about the low surface brightness universe and take advantage of the opportunity I am given by entering the team and having the chance to collaborate hand to hand with researchers.

Chapter 2

Data

El segundo capítulo se centra en exponer los datos utilizados para la detección de estas galaxias ultra débiles. Comienza exponiendo las características del Large Binocular Telescope (LBT), con el que se tomaron las imágenes, entrando en detalle en las prestaciones del telescopio así como de sus CCDs, sistemas de óptica adaptativa y espejos. Posteriormente se habla de la galaxia elegida para realizar las observaciones, que en este caso se trata de NGC1042. Más adelante se desarrolla la estrategia seguida para el proceso de toma de datos, desde la toma de las imágenes, pasando por la reducción de estas, astrometría y fotometría. Finalmente se introducen las galaxias ya conocidas dentro de la imagen, así como una galaxia descubierta recientemente por el estudio LIGHTS. Todas ellas tomarán un papel importante en nuestro trabajo, pues servirán como objetos de referencia en numerosos pasos del algoritmo para poder realizar bien las detecciones.

As explained in the Introduction, going very deep in the night sky has a lot of technical obstacles. Low surface brightness ($\mu > 30$ mag/arcsec^{-2}) is within the reach of today's large telescopes with a modest amount (few hours) of integration time. This chapter, we will talk about the technology and methodology that make this ultra-deep imaging possible.

2.1 Instrumentation and observation

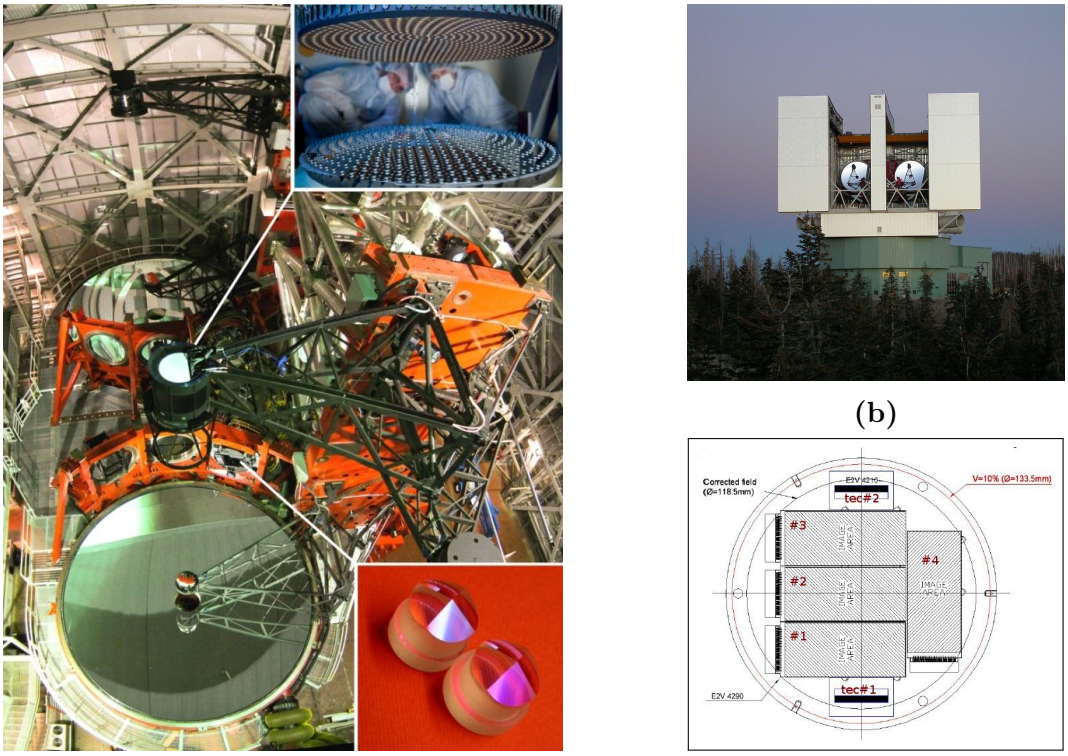
2.1.1 The Large Binocular Telescope

The images we will be using for this work were taken with the Large Binocular Telescope (LBT). The LBT, shown in Figure 2.1b, is one of the few binocular telescopes working right now and it is located in the Mount Graham National Observatory in Arizona, USA.

This telescope uses two 8.4 meters wide mirrors working together in the equivalent light-collecting ability of a 11.8m wide single mirror telescope and the spatial resolution of a 22.8m wide one. These two characteristics make this telescope a very good option for our observations.

It is also important to note that since 2003 there have been some studies to apply an FLAO[1] system on his secondary mirror to improve atmospheric aberration (see Figure 2.1a),

[1]The *First Light Adaptive Optics* is an optic system composed by an array of mirrors which can adapt



(a) **(b)** **(c)**

Fig. 2.1. Large Binocular Telescope parts. In (a) we can take a closer look on the adaptive optics system in the secondary right-side mirror of LBT. In (b) we have LBT's outside structure composed by a double-windowed structure. Finally in (c) we have a focal plane layout of the CCD for each LBC, this scheme belongs to LBC Red, LBC Blue follows the exact same layout.

achieving in certain wavelengths the Hubble Space Telescope sharpness. These studies were firstly proposed in 2003 by Esposito et al. (2003) and the FLAO system was finally implemented on 2010 by the same team (see Esposito et al. (2010)).

2.1.2 Observation target

The observation target for this project is NGC1042, which is a Sc type spiral galaxy located at R.A.(2000)=02h40m24.0s and Dec(2000)=-08d26m01s. Its distance to Earth is D = 13.5 ± 2.6 Mpc, which was recently measured using the Tully-Fisher relation (see Monelli & Trujillo (2019)). Furthermore, NGC1042 is located in a region of the sky with relatively low Galactic dust contamination, which are good conditions to study low surface brightness structures.

The NGC1042 region has recently attracted a lot of attention due to the claim that it contains two galaxies "lacking dark matter" (see van Dokkum et al. (2019)), but unfortunately these galaxies are not in the field-of-view of our image. In addition, there are some very diffuse galaxies that we will discuss later on the document.

their relative position in order to correct wave-front aberrations after the light propagation through our atmosphere.

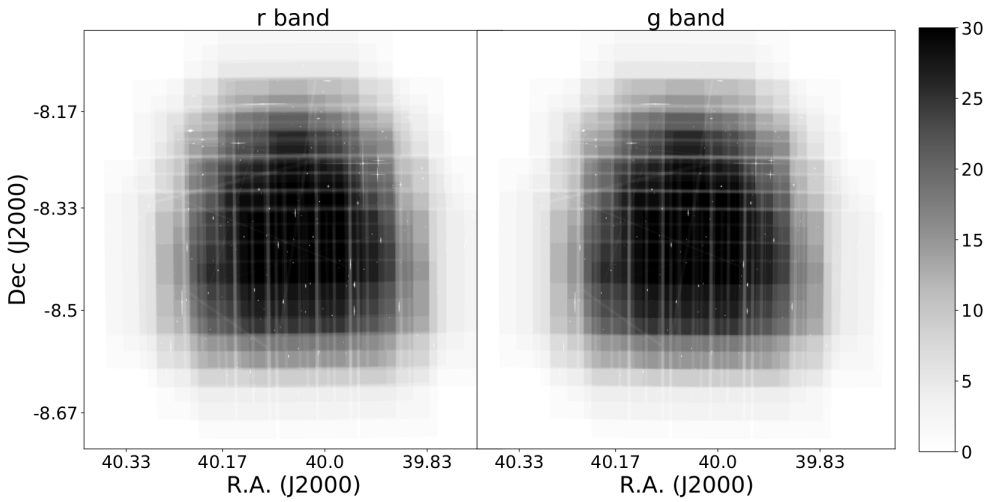


Fig. 2.2. Weight maps that show the stacking of images for both Sloan g-filter and r-filter. As seen in the grey bar, darker areas mean more stacked images per pixel.

2.1.3 Observation plan

The ultra-deep observation of NGC1042 and his surroundings took place in October 2020 using the LBT Large Binocular Cameras (LBC[2], Figure 2.1c). These are two wide-field cameras mounted on the prime arms of the LBT. Each of these cameras, called LBC Red and LBC Blue are optimized for a particular wavelength range, these are from 5500 Å to 1 μm in LBC Red and from 3500 to 6500 Å in LBC Blue. Both LBC cameras are composed of 4 CCDs each of which covers a size of 7.8 arcmin x 17.6 arcmin, with gaps between the chips of $\sim 18''$ giving a total field-of-view of 23 arcmin x 25 arcmin. CCDs layout in each camera can be found on Figure 2.1c.

In regards to the observation, SLOAN-g was used on LBC Blue camera and SLOAN-r on LBC Red camera. The total observation time on source was 1.5 hours for each filter split into 30 exposures of 180 seconds each. Being the short time granted for the observation, the flat-field imaging was obtained from the data itself instead of taking the flat images separately, as done in Trujillo & Fliri (2016). The images are also taken using a dithering pattern (which can be also seen at Figure 2.2) so that the galaxy never lies on the same region of the same CCD, this will also help reduce any possible instrumental noise.

2.2 Image reduction and photometry

2.2.1 Bias and flat correction

The procedure for the reduction of the image was executed as follows. First, a sigma-clipping median master-bias was made with all the bias images using `gnuastro` and subtracted from the rest of the data. Then master-flats for each filter were produced using the data itself, as we pointed earlier. After this, every image was divided by its corresponding masterflat depending on the filter.

[2] All the information related to the *Large Binocular Cameras* can be found at: http://abell.as.arizona.edu/~lbtsci/Instruments/LBC/lbc_description.html

2.2.2 Astrometry, sky subtraction and photometry

The astrometry of the single images was conducted using gaia eDR3 catalogue[3] to get a first astrometric reference that was later on improved using `SCAMP` (Bertin (2006)), which defines a distortion coefficient for each image. As we are using LBC cameras and each CCD has different distortion, this process is applied to every CCD separately. Later on, all 4 CCDs of each image (Figure 2.1c) are combined using a software called `SWarp` into a 9501×9501 pixel grid.

At this stage, sky is subtracted from the CCD exposures by masking the signal using `NoiseChisel` (Akhlaghi & Ichikawa (2015)). Then, a median of the remnant sky is subtracted from the image. Finally all these resulting images are combined using a median co-addition with `gnuastro arithmetic` program.

This final co-added image is deeper than the individual single exposures used, that is why it contains all these interesting faint objects in it. All these objects are affecting the sky image, and now that we know their location we can repeat the masking process of the individual exposures including these objects in the mask. That is how we get to the final co-added image.

In Figure 2.2, we can see the weight maps associated to each filter, in these two images there is a representation of the stacking process of each individual frame. We can see the dithering pattern used to image NGC1042 and its surroundings, centering the cameras on different locations for each exposure, this, as explained on Trujillo & Fliri (2016), helps reducing image noise in the stacking process.

The photometric calibration is performed using SDSS DR12 images (see Alam et al. (2015)). For this calibration, around 600 stars of magnitudes between 18 to 22 for both filters were used. Then the match is made and these zeropoints are obtained:

$$ZP_g = 34.527 \pm 0.006 \pm 0.01 \; mag$$
$$ZP_r = 34.111 \pm 0.006 \pm 0.01 \; mag$$

The errors following the values are the statistical error and the typical photometric zeropoint error stated by SDSS respectively.

2.3 Final image

In Figure 2.3 we can see the final color image composed by the two Sloan r and g filters obtained after all the reduction, sky subtraction and photometric calibration. If we take a look inside the central region in which the pixels have 24 or more exposures each, the faintest surface brightness achieved is 31.2 $mag/arcsec^2$ for g-filter and 30.5 $mag/arcsec^2$ for r-filter.

2.3.1 Analysis

Now that we have the final science image (Figure 2.3), we can focus on the analysis and detection of the faint galaxies we introduced previously in this document. This will serve

[3] All information about Gaia *Early Data Release 3* can be found at: https://www.cosmos.esa.int/web/gaia/earlydr3.

as an introduction of the LIGHTS survey (submitted, Trujillo et al. (2021)) potential for detecting these low surface brightness (satellite) galaxies.

We will use these 5 objects as a reference sample of what LIGHTS survey is capable of detecting:

- **SDSSJ024007.01-081344.4** - An example of a relatively bright object.
- **NGC1052-DF1** - The faintest low surface brightness galaxy in the field-of-view discussed in Cohen et al. (2018).
- **T20-12000** & **LSB21** - Two low surface brightness galaxies present on deep surveys, spotted in Tanoglidis et al. (2021) and (Román et al. 2021, in prep), respectively.
- **LBT1** - A galaxy not previously reported, dubbed as LBT1. (submitted, Trujillo et al. (2021)).

The first four of these objects (excepting LBT1) have been detected by the other shallower surveys (SDSS, Dragonfly), but LBT imaging helps clarify the real nature behind them and detects more accurately the foreground and background sources that are contaminating their detection, which will help measuring better the properties of these objects. This can be seen clearly in the comparative figure of these objects between the rest of the surveys and the LBT shown in Figure 2.4.

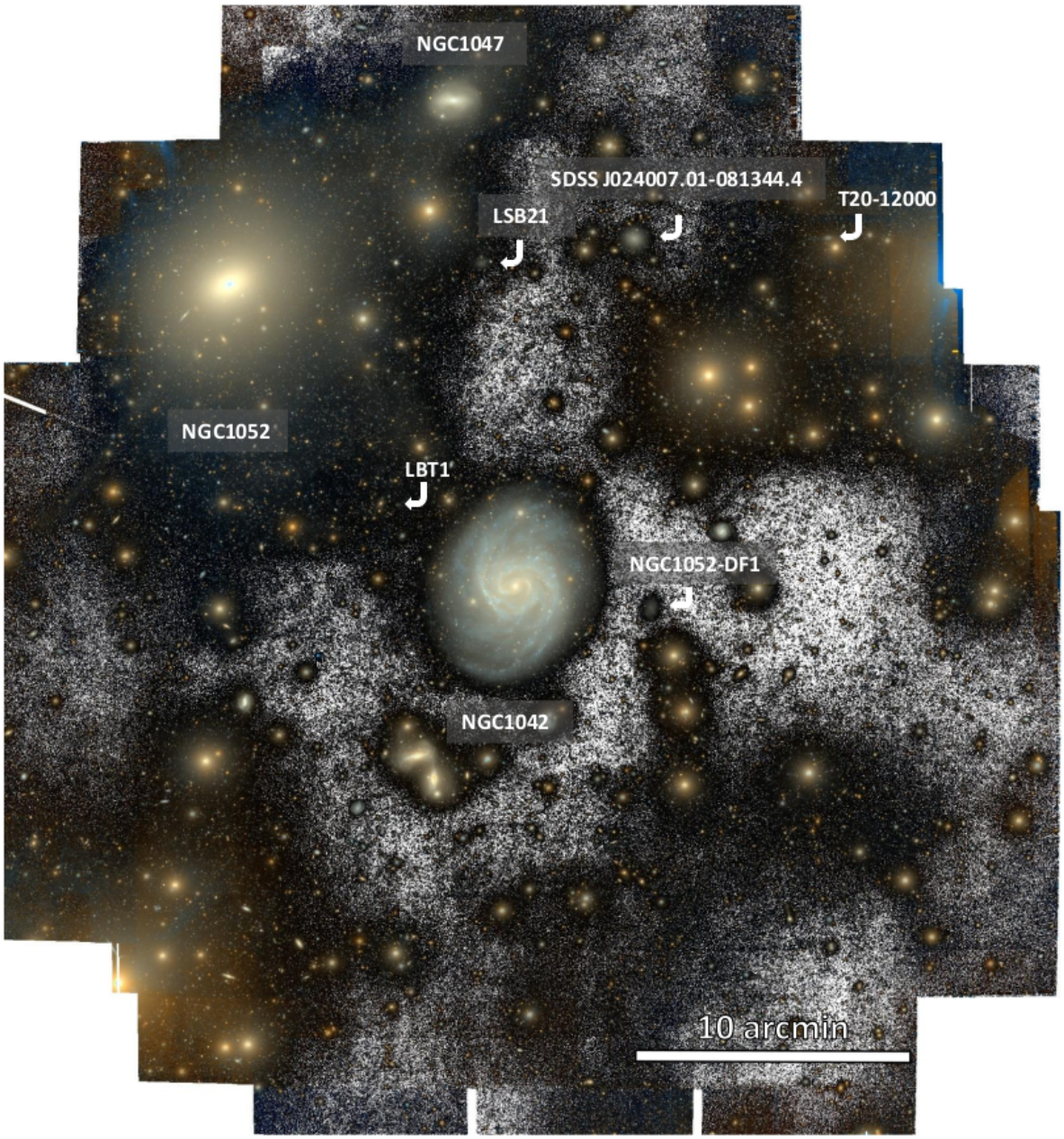


Fig. 2.3. Final composition of the NGC1042 and its surroundings image taken by LBT. We can see NGC1042 in the center and all these faint features, like LSB21, DF1 or the new identified galaxy (LBT1).

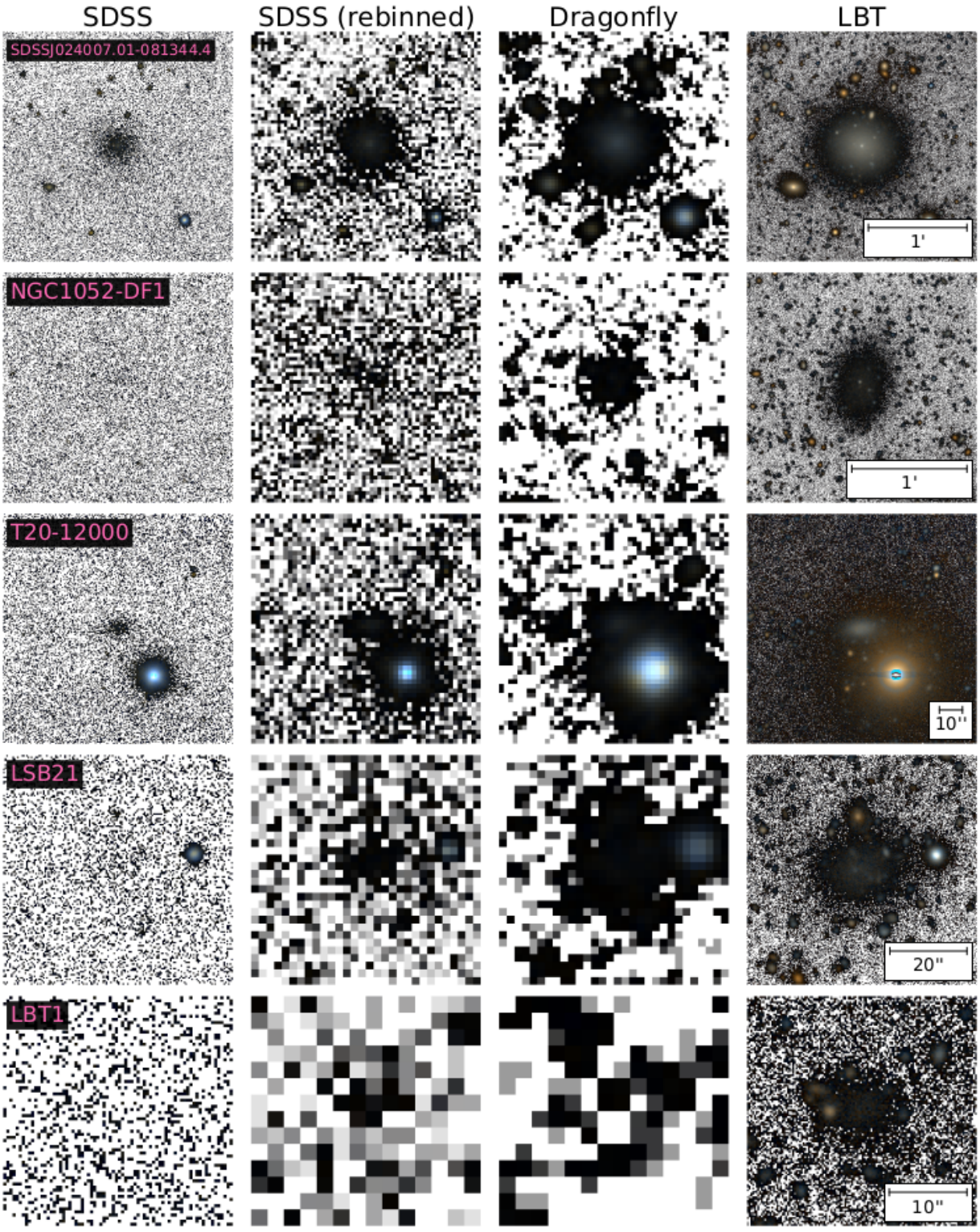


Fig. 2.4. Five low surface brightness galaxies in the surroundings of NGC1042. The columns show SDSS survey, SDSS rebinned to fit Dragonfly pixel scale,Dragonfly and LBT.

Chapter 3

Detections

En este tercer capítulo se explica paso a paso el proceso seguido para detectar los objetos de interés utilizando los datos presentados en el capítulo anterior. Primeramente introduciremos las herramientas que vamos a utilizar a lo largo del algoritmo, las cuales forman parte del paquete ***gnuastro****. Seguidamente comenzaremos con las detecciones, las cuales consisten en obtener la parte deseada de la imagen inicial y separar la señal del ruido dentro de la imagen para posteriormente dividir esta señal en objetos que recogeremos en un catálogo. Después analizaremos este catálogo para extraer de él un catálogo más acotado con posibles galaxias satélite, que examinaremos finalmente para llegar a los objetos definitivos. Todo este proceso se explica de forma minuciosa y se acompaña de dos Apéndices con explicaciones complementarias.*

In this chapter, we will discuss the algorithm for detecting these faint galaxies step by step.[1] We will start introducing the `gnuastro` tools we will be using and then we will jump into the detection process, which starts by making a weighted image, following by a separation of signal and noise, the detection and catalogue making of our desired objects and ending with the catalogue analysis.

3.1 Introduction to `gnuastro` tools

In this section, we will make an introduction of the `gnuastro` tools we will be using in our algorithm. Since we will be describing their usage in a more detailed way on next sections, we will just enumerate the tools and give some clues on their usage and power.

- **Arithmetic:** Used to operate on any number of images. It is a large number of mathematical operators as well as some statistical operators too, like `median, average, min, max`.
- **Crop:** It crops a region or regions for an image with the input of center and crop width.
- **Fits:** It gives information and helps manipulating FITS multi-extension files as well as headers.

[1] All the code scripts and pipeline used in this work can be publicly accessed in: https://gitlab.com/AndyGarciaSerra/trabajo-de-fin-de-grado

- **MakeCatalog:** It makes a catalogue of a labeled image with the different measurements you need.
- **Makeprofiles:** It creates mock 2D profiles in a FITS image using Monte Carlo integration.
- **Match:** It matches catalogue objects within a given aperture between coordinates.
- **Noise Chisel:** It detects signal and separates it from noise. (see Akhlaghi & Ichikawa (2015))
- **Query:** Interface to query external data sets and download the required catalogue of objects within an image.
- **Segment:** It segments detected regions based on the signal structure in the input.
- **Statistics:** It makes statistical calculations and gives some statistical information about any table column or image.

It goes without saying that each and everyone of these shell commands are highly adaptable and let you choose and change a large number of parameters, making them suitable for any situation regarding operating on astronomical data.

Now that we have introduced the tools we are going to use. In the next sections we will discuss the process of detection and parameter tweaking performed.

3.2 Weighted image

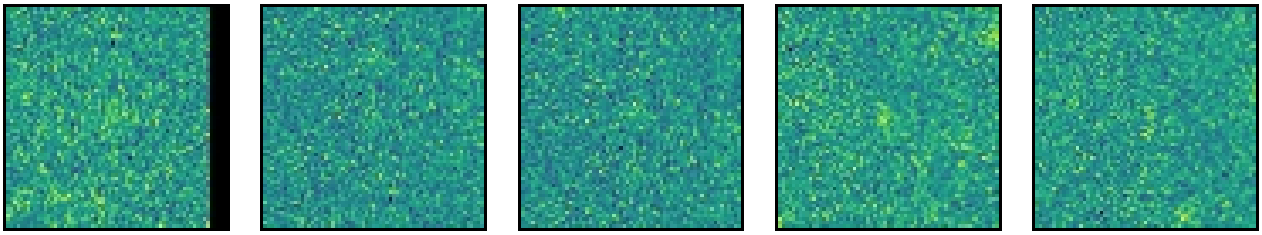
In this step, we simply take the weighted images we introduced earlier (see Figure 2.2, chapter 2) and make sure we are only using the pixels in our image that have 15 or more exposures. For doing this we will use the command `astarithmetic`.

In the early stages of the algorithm we did not take into account this first step, which made us to detect some false positives (see Figure 3.1) directly linked to the noise located at the edges of the image. In these edges, the pixels after the co addition of the single exposures have been made using between 0 and 10 single frames, as we can see in Figure 2.2. Figure 3.1a shows how this noise is commonly located around the borders of the image, as we can see a black region at the right corresponding to the image frontier.

For our case we have set a threshold of 15 exposures per pixel, which seems to enclose visually all the important faint objects and omits all the external noise we just showed. Also, 4 of the 5 objects we discussed as extended low surface brightness sources on Chapter 2 are inside this region. In Figure 3.2 we have the original image on the left and the deeper, weighted image with only the pixels corresponding to a composition of 15 stacked images at least on the right.

Now that we have our weighted image we do not have to worry about false positives in the form of noise, or at least we have omitted with a simple step a lot of them. We can then

[2] `ds9` is a image viewing software commonly used in astronomy.



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

Fig. 3.1. Examples of noisy crops classified as false positives when trying to detect extended faint sources. These crops were extracted from `ds9`[2]using a *viridis* color map. The crops have a size of 15 arcsec.

jump to the next step.

It is important to point out that from this first step onward we will just focus on one of both filters (r in our case) just because it makes the process easier. We will make every step with the r-filter and then for any information needed from the g-filter we can just extract it from the original g-filter, image by using the coordinates of the objects we have achieved via the r-filter.

3.3 Sky and signal categorization

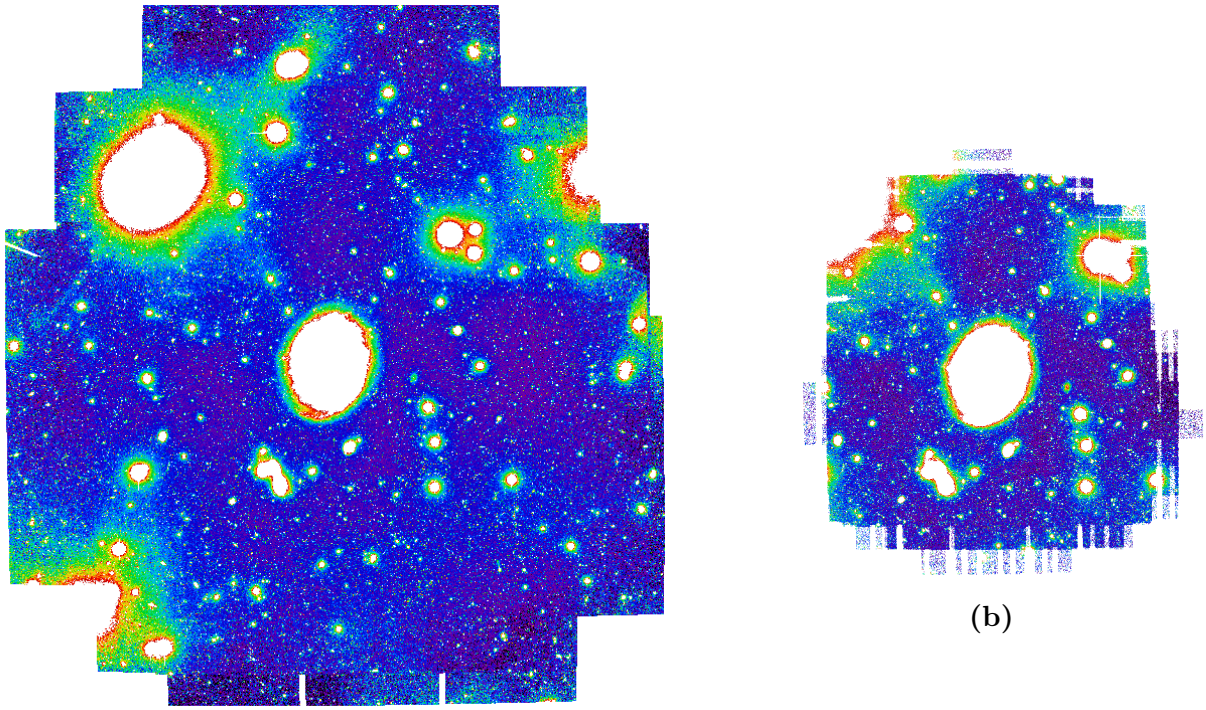
In this step, we will focus in the detection of signal versus noise in our main images. This is an important step when working on images that have big gradients coming from big sources, like NGC1052 in our case (see Figure 2.3).

All these light gradients are normally treated as noise because of how `NoiseChisel` steps work. `NoiseChisel` separates pixels in tiles such as each tile is enclosing a concrete number of pixels. These tiles are then compared with the surrounding tiles. The algorithm compares the mean and the median in such way that if a section is clearly uniform or close to, then it is not treated as signal. This is why it is important to tweak these parameters for the gradients to be included in the signal.

For our project, this step is not as important being that we are not interested in galaxy wings or light gradient coming from galaxies. Anyway, the process of separating signal and noise is treated in depth in Appendix B.

Tweaking some of `NoiseChisel` parameters, we will focus on making a good sky image so our signal is well established. After applying `NoiseChisel` our detections of the noise and signal are very promising, having an almost flat sky with very low standard deviation map. Then we can just make sure our objects of interest are inside the signal extension of the FITS file created by `NoiseChisel`. The detection map is shown in Figure 3.3, it consists of a FITS file with pixel values 1 at the signal and 0 at the sky.

Now that we have isolated the signal and the noise we can go to the next step, which is making a segmentation of the signal using `Segment`.



(a) (b)

Fig. 3.2. Science image of NGC1042 and its surroundings before (a) and after (b) applying the weighted region step of the algorithm. These images were created using `ds9` with `sls` color map and `zscale` contrast scale.

3.4 Signal segmentation

This is a vital step regarding the detection of our low surface brightness galaxies. To make an easier analysis of the detections, we will focus on one of the objects shown before in chapter 2. We will be using LBT1 (see Figure 2.4, last row) as a reference, being the rest of them too big or just too affected by the light coming from foreground or background sources close to them. This last case is the case of DF1, T20 and LSB21 (see Figure 2.4, rows 2, 3 and 4 respectively).

The `gnuastro Segment` program uses a detection image as an input (something like the `NoiseChisel` output we saw in Figure 3.3) to make a segmentation of these detected regions and split this detection into clumps, which are labelled separately and can be then turned into catalogue of interest objects.

The first step is called convolution, and it is basically a smoothing algorithm based on a given kernel. This step will help highlighting the regions of the image we would like to be detected as clumps to make our catalogue. That is why it is important that this convolution kernel fits our needs helping `segment` detect the objects needed.

In Figure 3.4, first column, there is a comparison figure between the default convolution kernel (second row) and the custom kernel we used (first row). We can clearly see how the different kernels highlight different regions of the image as needed.

In our case, we used a Gaussian shaped kernel with 5 pixels of FWHM and truncated at 3 times the FWHM. This detected very well our objects making clumps where the diffuse

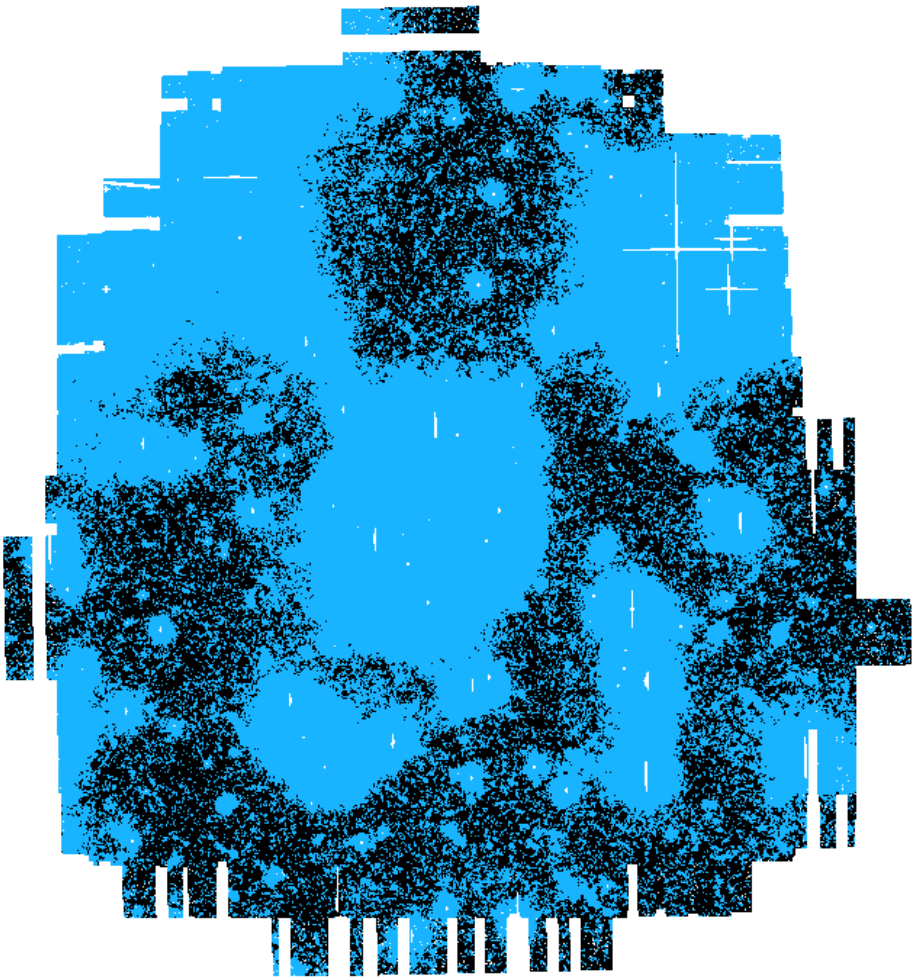


Fig. 3.3. Detection map output of `NoiseChisel`. In blue we can see the pixels treated as detections and in black, we can see the pixels considered sky or noise. Image created using `ds9` with `cool` colormap.

regions were. We tried in previous stages of the algorithm with different kernels until we got this one that is working flawlessly with our expectations.

Then after the convoluted image is segmented, we can see again in Figure 3.4 how the central clump of the image is maintained as a final clump in the case of our kernel. That is not the case using the default kernel.

For a deeper analysis on how to choose properly the convolution kernel go to Appendix A, where we explain how the `segment` algorithm works step-by-step showing these kernel examples and how they influence the final clump detection.

Finally, making a visual inspection on the segmented image over the original data we can identify some potential objects and see their segmented clumps to verify the kernel was applied properly. This is the case of the four objects shown in Figure 3.5, where we are showing in the first row the original data crops for these objects and in the second row the segmented clumps. This helped us confirm that the segmentation was good enough for our needs and we could jump to the next step, the catalogue making.

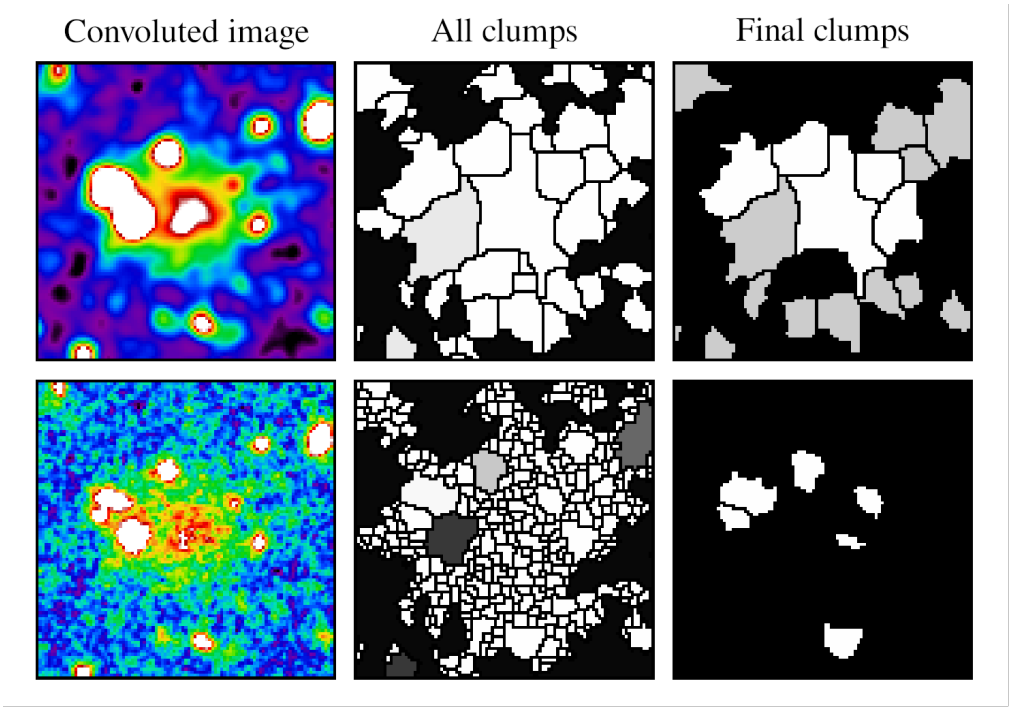


Fig. 3.4. Figure showing the clump detection made by `segment` for different convoluted images. First row shows a custom kernel convoluted image while the second row is showing the default convolution of `segment`. In the columns, we can see from left to right, the convoluted image, all the intermediate clumps and the final clumps.

3.5 Making a catalogue of detections

At this point of the algorithm, we have divided signal from noise, detected the objects we need for our catalogue and labelled them. Now we need to make a catalogue of these objects so we can later on make a visual inspection and classification of them.

Most existing common tools for astronomical data-analysis (for example `SExtractor`) merge the two processes of detection and measurement (catalogue production) in one program. However, in `gnuastro` detection is separated from measurements and catalog production. This modularity is key for a full personalization of the data set you want to create.

First, we will introduce `MakeCatalog`[3] measurements we will later on use on our catalogue to analyze the detections, then we will look at the different plots on the initial detections and analyze the different sources detected. At that point, we could start over-plotting our objects of interest on top of our detected objects and finally we will try to make an object selection that enclose the objects we are interested in, which we will later on inspect to reach a final sample of possible objects.

3.5.1 Catalog measurements

These are some of the `MakeCatalog` possible measurements, we will describe them as the command you need to write in shell to call the program:

[3] https://www.gnu.org/software/gnuastro/manual/html_node/MakeCatalog.html

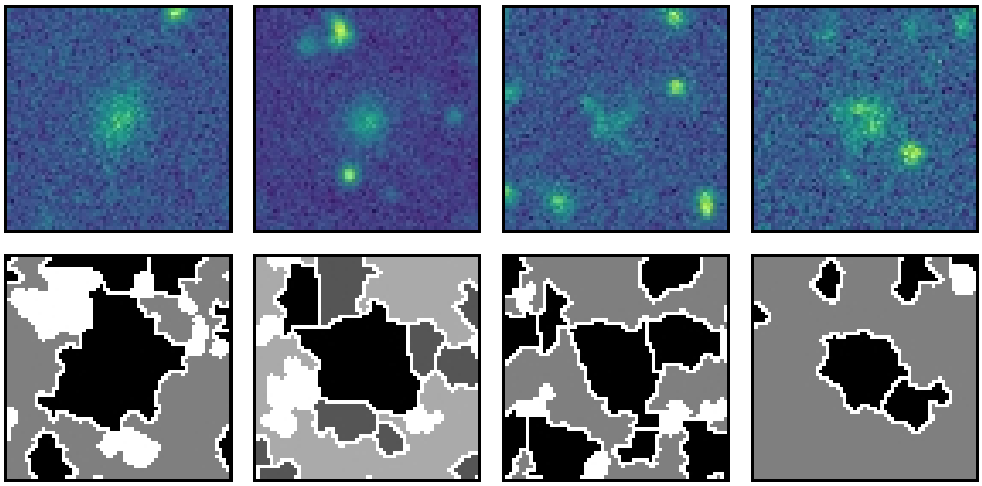


Fig. 3.5. Some examples of objects detected as clumps by `Segment` with our current kernel. The first row is showing the objects in a `Viridis` color map and the second row is showing the clumps extension centered to the current object so that the object clump is appreciated. The crops have a size of 15 arcsec.

- `--ids` : It gives an identification number for each object and another one for each clump inside that object.
- `--ra` : It extracts the Right Ascension of the object mapped with the original data image.
- `--dec` : Same that `--dec` but with the object or clump declination.
- `--magnitude` : It calculates the magnitude of the labeled clumps or objects.
- `--sn` : It calculates the signal to noise ratio of all clumps or objects. (See Akhlaghi & Ichikawa (2015) for the exact equations used.)
- `--halfmaxradius` : Radius containing half the maximum flux of the labeled region.
- `--halfmaxsb` : Surface brightness ($magarcsec^{-2}$) within the region that contains half the maximum value of the labeled region.
- `--halfsumradius` : Radius containing half the total sum of pixels of the labeled region. Also known as the effective Radius.
- `--halfsumsb` : Surface brightness ($magarcsec^{-2}$) within the region that contains half the total sum of pixels of the labeled region. This is also known as effective surface brightness ($\langle\mu\rangle_e$).

These measurements are a small sample of all the possible measurements introduced by `MakeCatalog`. Some of these are the ones used by the algorithm to make the catalogues and some that do not have a high impact in the catalogue analysis.

Once we have introduced the terms, we will make our catalogue, which will include identifications, RA and DEC, magnitude, radius at half sum and at half maximum and surface brightness at half sum and at half maximum too.

3.6 Catalogue analysis

In this section, we will focus on the analysis of the data included in our catalogue, plotting the different measurements we have just introduced. It is important to point out that all the following plots are made using the r-filter catalogue of detections. As we said earlier in this document focusing on one filter is an easier way to make the analysis of data. Then, if we need to make a comparison between both catalogues or we need the usage of some measurements on the g-filter catalogue we will make it and extract the information we need from it.

In this regard, one of the most important measurements not mentioned above is the color, which we can calculate by subtracting the r and the g magnitude. This color can be a good indicator and can help us identify some of the objects in our catalogue. For example, redder galaxies tend to be more red-shifted.

In order to have something as a reference to compare with we can over-plot on top of our objects all the Gaia[4] sources that are within our image. To do this, we will be using `gnuastro`'s `query` command. This program allow us to check in the Gaia database (in our case we chose the last data release, edr3) for any objects in the field-of-view of our image and makes a table with these objects coordinates and data. Then, to make things easier, we matched this catalogue with our catalogue of objects using the `Match` program and finally we could see the number of sources from both catalogues. Whenever we plot a measurement from the catalogue we will be plotting the Gaia sources too to make a comparison of the behavior of these and understand better where the location of our desired objects is.

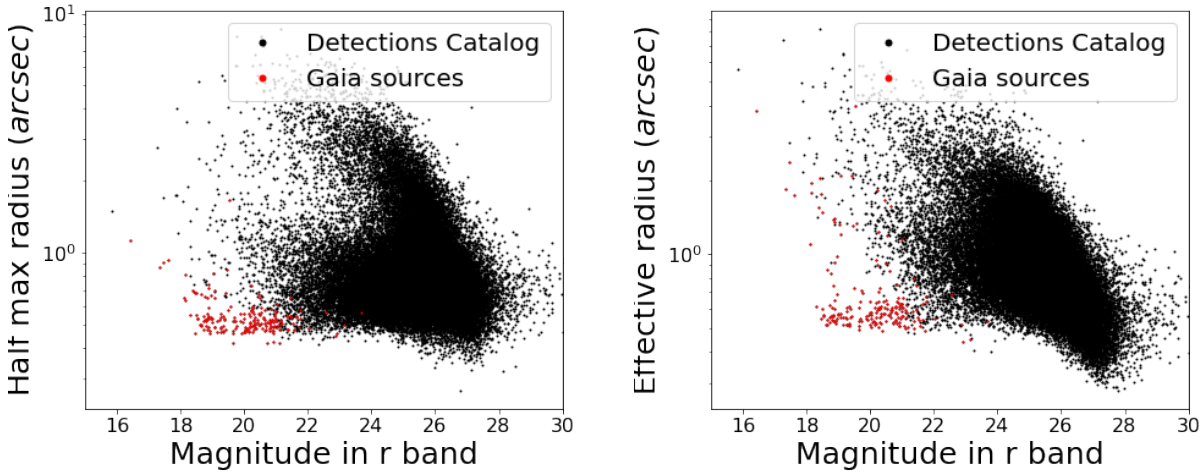
3.6.1 Effective radius and half max radius

First we can start by taking a closer look at the plots of effective radius and the radius at half the maximum pixel value. These plots will help us clarify the difference between them, which can seem very similar and confusing. In our case, the difference between these two measurements will play a very important role when selecting and differentiating our objects, as we will see later on this chapter.

In Figure 3.6, we can see these two measurements plotted side-by-side. We can clearly see their similarity in shape, but we can also take a closer look on the differences. Firstly, we can notice how for the half max radius (left side, 3.6a) not only for the highest values but for the main black trend shape the values of the radius are higher than the ones in the effective radius plot (right side, 3.6b). This is caused because half max radius measurement does only care about the highest pixel value inside the clump. As we used a wide custom kernel instead of a sharp one to make our detections, the majority of our detected objects will not be as sharp, this means that the brightest pixel of each object will not be much brighter than the others, making the half max radius larger than the current effective radius for the majority of objects.

Now we can watch closely the Gaia sources behavior and we can see again a very big difference between both plots. For the half max radius the Gaia sources seem to be restrained to a range of radius, not being higher than 5 pixels approximately whereas for the effective radius this changes, having objects that achieve even 20 pixels of radius. This is caused by

[4] https://gea.esac.esa.int/archive/



(a) (b)

Fig. 3.6. Plots representing the half max radius (a) and the half sum radius (b) of our detection catalog against the magnitude in Sloan r. Red dots correspond to all Gaia sources within the field-of-view.

the opposite phenomenon of which we explained in last paragraph. For point-like profiles, which is the case of gaia sources, as the brightest pixel is much brighter than the actual pixels around it, the half maximum is higher, meaning a thinner radius than the effective radius, which has to contain much pixels around the peak which correspond to half the total sum of pixels.

It is also clear that for the Gaia sources the radius trend to be much lower than for the objects we are trying to detect. This is obvious knowing that Gaia objects have a point-like profile in general, being even very similar in numbers one another.

3.6.2 Effective surface brightness and half max surface brightness

Now that we have established the difference between the above radius we can move to the next measurements of interest, which are the surface brightness at half maximum and the effective surface brightness.

To make a better comparison of the data, we will plot now four of the low surface brightness objects introduced in Chapter 1. The fifth object was not plotted because it is out of the field-of-view of our weighted image, it laid in the region where pixels have less than 15 exposures so the object got rejected by our first step (see 3.2 for more details). Again, we will show the Gaia sources in the plots to see the different tendencies of sources and to decide which region of the plots has a higher density of these low surface brightness objects we are looking for.

In Figure 3.7 we have the mentioned plots. We have made a second row zooming into the region of interest for an easier visual inspection. First, we will take a look at the shape of the Gaia sources in both plots. We can see how they follow an almost straight line in both cases, this is caused by the shape of Gaia sources, which are mostly (non saturated) stars. This gives us the clue that stars will follow this line and if we continued this line to fainter

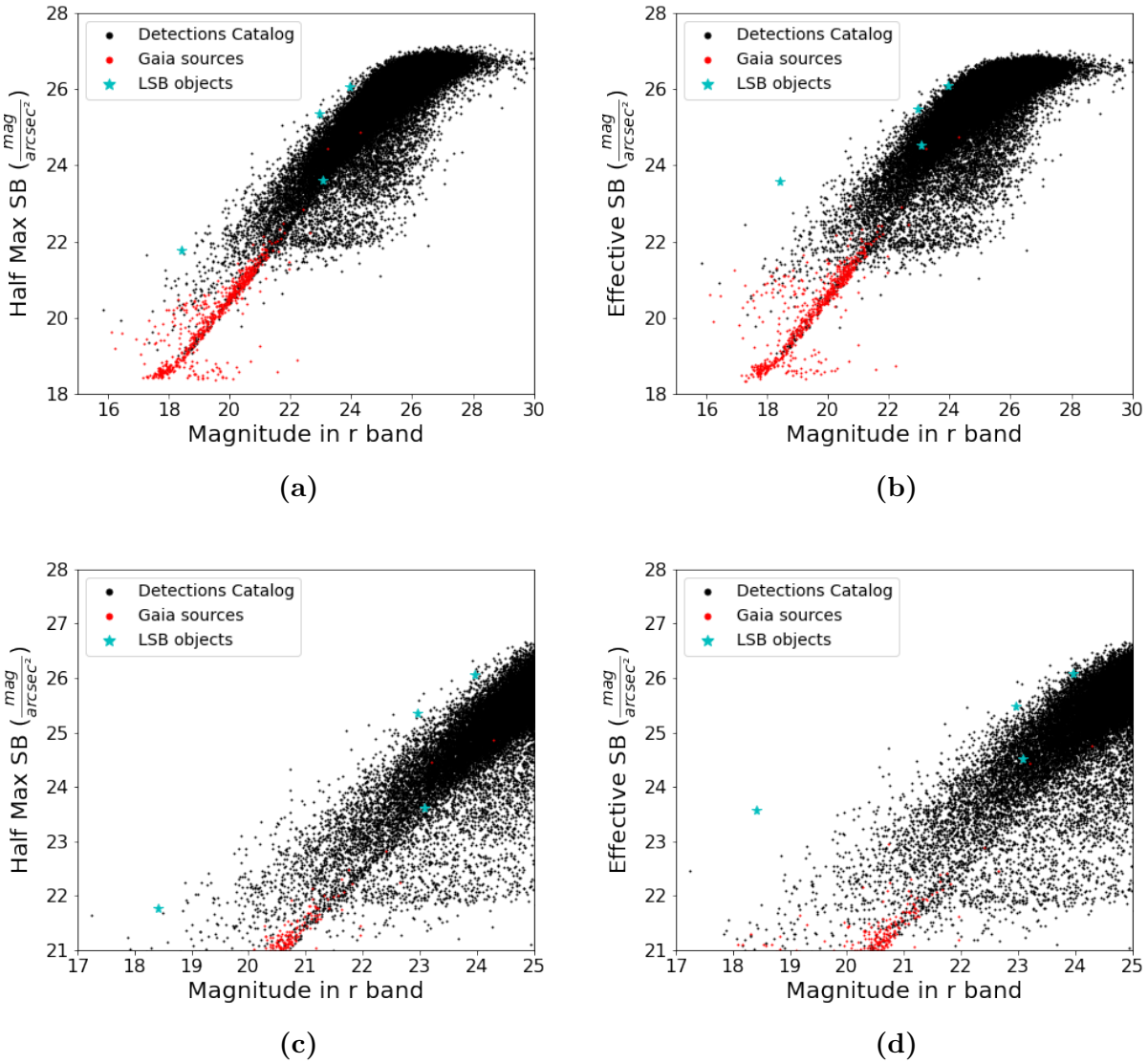


Fig. 3.7. Plots representing the half max surface brightness (a, zoomed in c) and the effective surface brightness (b, zoomed in d) of our detection catalog against the magnitude in Sloan r.

values (larger magnitude) these will be very faint stars.

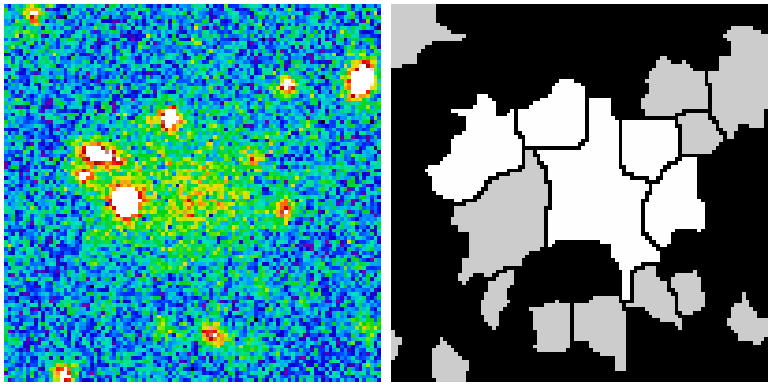
We can now focus on the LSB objects we represented in cyan color with star shapes. We can see in both measurements how there is a point very deep into the black trend of the rest catalogue objects, at around magnitude 23. This object has obviously lower values of these measurements. These lower values, which means it is measured as brighter than it actually is, is caused by the contaminated light coming from sources at the foreground and background. This object is *DF1*, the first proposed low surface brightness object by the Dragonfly survey (Cohen et al. (2018)). We showed this plot in Figure 2.4 (second row) back in Chapter 2. In this representation without zoom the sources we are talking about are not easy to see, but if you look to the zoomed plots you can clearly see them.

The other object remaining at the left part of the plot has very different values between the two measurements. This object is *SDSSJ024007.01-081344.4*, a relatively bright object

we reviewed also in Chapter 2 (Figure 2.4, first row). As we can see in this Figure, this object has a very bright pixel in the center, which was detected in our catalog, making the profile of the object much sharper in the center and therefore the half max radius smaller than the effective radius. This makes the half max surface brightness higher than the effective surface brightness.

The goal to plot these measurements was to see a trend followed by these objects that helped us focus in a more specific region of the plots to find more of these objects. And we can see this clearly with the two objects remaining, even though they seem a very small sample.

At this point, using half max surface brightness is going to be a better indicator than using effective surface brightness. This is caused by how `segment` splits the data into clumps. In Figure A.2, we showed how the clumps were made around *LBT1* in comparison with the different convolution kernels, now, to see better the original image we will take a look at Figure 3.8, where we are comparing to the original image instead.



(a) **(b)**

Fig. 3.8. Images of LBT1 (a) and its clump segmentation (b). These where made using `ds9`, (a) was made using `sls` color map and `zscale` contrast scale, (b) was made with a customized contrast scale.

We can see how the segmentation here separates these clumps making the central clump smaller than what this object should really look like. Take in account that if we masked off the sharp clumps surrounding these diffuse region that are contaminating with light, this object will be the size of all these clumps together, more or less as the shape of we can see in the right side surrounded by black noise.

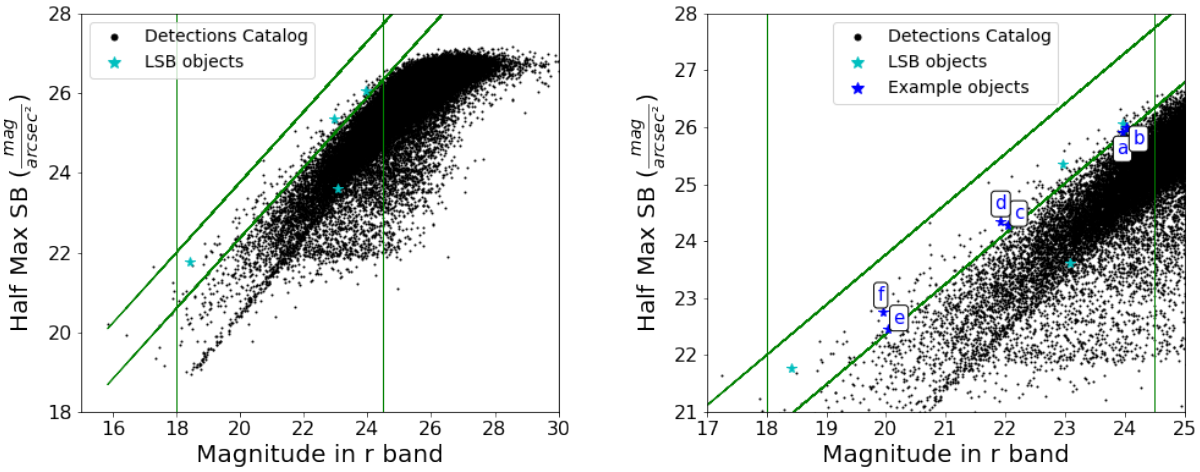
Being that this truncation of the central clump is so small, it can not hold all the sum of pixels, so here using the effective radius as a detection criteria will not be considering the hole sum of the real object we are trying to detect. That is why we introduced the use of Half Max instead as a better indicator, which only takes account of the brightest pixel in the clump as a reference. At this point we can not forget the physical significance behind the effective surface brightness which is used more often in the field, but for our case of study the Half Sum surface brightness is clearly a better indicator.

This matches with the shape of the plots shown before in Figure 3.7, in which we can tell that for the effective surface brightness differentiating between our objects and the main

black trend of the catalogue is very difficult, whereas for the Half Max surface brightness in the region of the two fainter cyan objects we can see a plateau clearly differentiating a less dense region of catalogue elements. There is where we will focus our attention from now on.

3.6.3 Catalogue selection and analysis

So once we know the region to focus on we will make a cut around it and inspect the objects inside. In Figure 3.9a we can see the bounded selection of the catalogue we will inspect. This region holds 208 objects of our catalogue that we can easily inspect by eye.



(a) **(b)**

Fig. 3.9. Catalog plots showing the interest region for the first inspection. In (a) we can see the region of the catalogue that we are going to examine. (b) shows a zoom of the left panel indicating with a tag the objects we are going to inspect closely.

As a first step to see the behavior of the detections inside this region, we have looked at 6 objects located at 3 different magnitude ranges, two at around 24 mag, another two at around 22 mag and the last two at around 20 mag. These objects are tagged and shown in Figure 3.9, labelled from (a) to (f) and their stamps are presented in Figure 3.10 with their corresponding measurements in Table 3.1.

Taking a look at the morphology of these objects we can see that (a) and (b) have a very diffuse light distribution, which indicates a good selection criteria. These two objects are also the faintest ones in magnitude and half max surface brightness, that is why they look the faintest also. Both seem to be low surface brightness galaxies, even looking at the effective surface brightness, which is very faint, at around 26 mag arcsec^{-2}, everything seems to confirm that they are the kind of object we want to be detected.

For the rest of the objects we can see how the center is much sharper and the general structure appears to be the one of a red-shifted galaxy. The fact that these last objects seem redder too gives us the clue that they might be distant spiral galaxies. These four objects are in the brighter part of the selection so we might take a look at similar objects in the entire sample to see their measurements also.

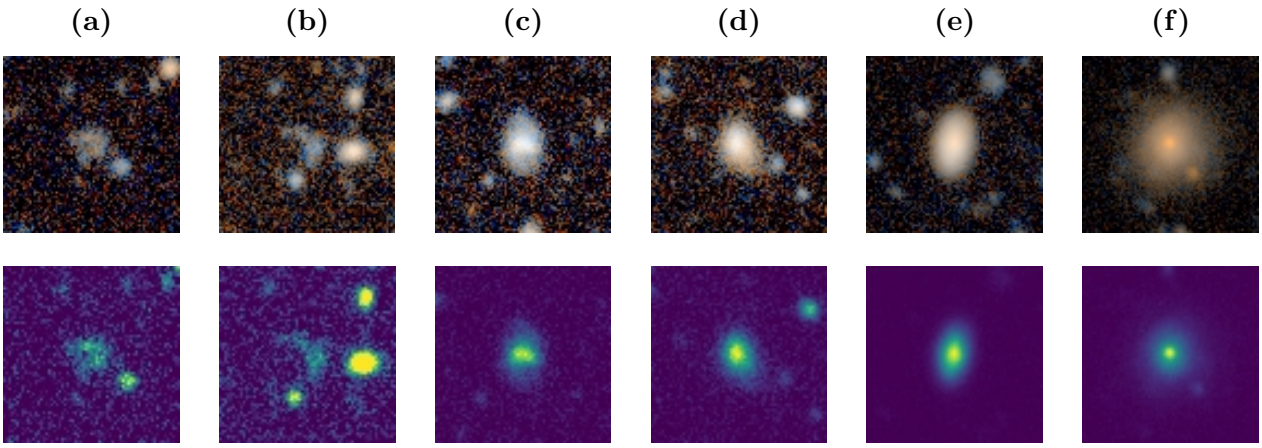


Fig. 3.10. Objects of interest at different r-filter magnitudes inside the catalogue selection made. First row corresponds to the rgb images and second row to the `viridis` color map images using `ds9`. Color images were made using a script developed by Raul Infante that will be a part of `gnuastro` shortly. These crops have a size of 18 arcsec.

Table 3.1. Six examples of objects inside the selected region of the catalogue. The table include the label of the object, effective surface brightness, Half Max surface brightness and magnitude, all three in both g and r filters.

Object	Eff SB (r)	Eff SB (g)	SB_{hm} (r)	SB_{hm} (g)	m_r	m_g
	$(mag/''^2)$	$(mag/''^2)$	$(mag/''^2)$	$(mag/''^2)$	(mag)	(mag)
(a)	25.94	26.41	25.92	26.49	23.97	24.38
(b)	26.01	26.31	26.00	26.17	24.03	24.13
(c)	24.51	24.94	24.29	24.71	22.04	22.34
(d)	24.53	25.04	24.36	24.83	21.93	22.50
(e)	22.53	23.10	22.45	23.17	20.02	20.67
(f)	23.54	24.59	22.75	24.18	19.96	20.97

Having a look at the 208 objects of the sample we can differentiate roughly three types of objects. The first ones, which are the least interesting, are noise and unwanted artifacts, like galaxy wings. The second kind of objects are spiral galaxies, with noticeable sharp nucleus, there are face-on and edge-on ones. The last type of objects are the ones we are trying to find, which are very similar to objects (a) and (b).

To have a clearer perception of how these objects are at the different parts of this selected region, we will now classify the objects we are sure have a sharp center that could be spiral galaxies and the diffuse ones that are LSBG candidates. The rest of the objects that do not follow either of these standards will not be classified.

Once we have made that differentiation, we plot these objects again in a Half Max surface brightness against magnitude plot, which is shown in Figure 3.11. If we take a look of the distribution of the sources we can differentiate three regions. The first one is between 21 and 23 mag arcsec^{-2}, in this first bright region almost every object is classified as a sharp centered one, being the majority of them high redshift galaxies. The second region is the intermediate one, from 23 to 25 mag arcsec^{-2}, and holds up various objects, some of them are still sharp in the center but there are also some diffuse ones as we go fainter following the downside

boundary; this region is an overlap region where these two types of objects coexist. Finally we have the last and faintest region, from 25 to 26.5 mag arcsec^{-2} approximately, where the majority if not all of the objects are the ones classified as diffuse and LSBG candidates. This is the part where we will have to look at closely.

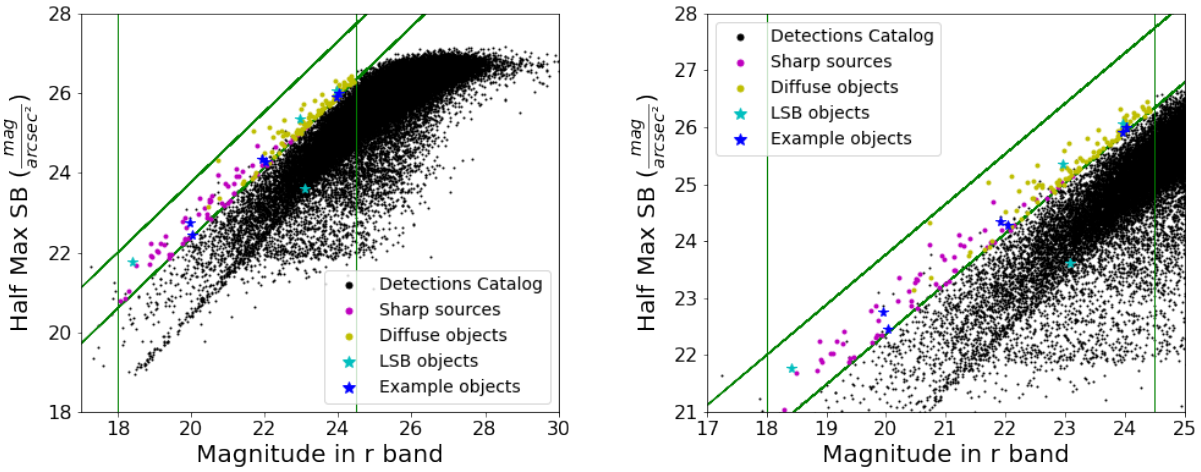


Fig. 3.11. Plots showing the location of sharp objects versus diffuse ones in a Half Max SB against magnitude plot. The right plot is a zoomed version of the selected region for a better visual inspection.

Chapter 4

Results and discussion

En el cuarto capítulo presentaremos los ejemplos definitivos de galaxias de bajo brillo encontrados por el algoritmo, analizando y comparando estos resultados con las galaxias ya conocidas. Además, calcularemos las magnitudes de estas galaxias y se presentan los datos del catálogo en una tabla para después poder colocar en una gráfica estas y ver si siguen la tendencia que habíamos tratado de encontrar en el capítulo anterior.

Once the final region of interest is clear, 6 of these low surface brightness objects are represented in Figure 4.1, which look clearly similar to the ones in the literature we showed back in chapter 2. The location of these objects around NGC1042 is shown in Figure 4.2.

Looking at the images it is clearly noticeable how satellites 2, 4 and 6 are the faintest, having an effective surface brightness between 26 and 26.6 mag arcsec^{-2} for the g-band. This is very faint and similar to other galaxies named before in the literature such as LBT1 (See Figure 3.10, submitted Trujillo et al. (2021)), which has a central surface brightness of $\mu_r(0) = 26.41$ mag arcsec^{-2}.

Following the criteria shown by Martin et al. (2019) we can classify low surface brightness galaxies by their effective surface brightness ($\langle\mu\rangle_e$) in three different categories:

- *High Surface Brightness Galaxies* (HSBGs): Defined as galaxies with $\langle\mu\rangle_e < 23$ mag arcsec^{-2} in the r-band. They represent the vast majority of galaxies in past surveys, such as SDSS.
- *Classical Low Surface Brightness Galaxies* (Cl.LSBGs): Defined as galaxies with $23 < \langle\mu\rangle_e < 24.5$ mag arcsec^{-2} in the r-band.
- *Ultra Diffuse Galaxies* (UDGs): Defined as galaxies with $\langle\mu\rangle_e > 24.5$ mag arcsec^{-2} in the r-band.

Anyway, there is not a clear definition in the literature of what constitutes a UDG. That is why 'Low Surface Brightness Galaxy' (LSBG) will be used to refer to any of the satellites found with an effective surface brightness fainter than 23 mag arcsec^{-2} in the r-band. This is the case for each of our objects.

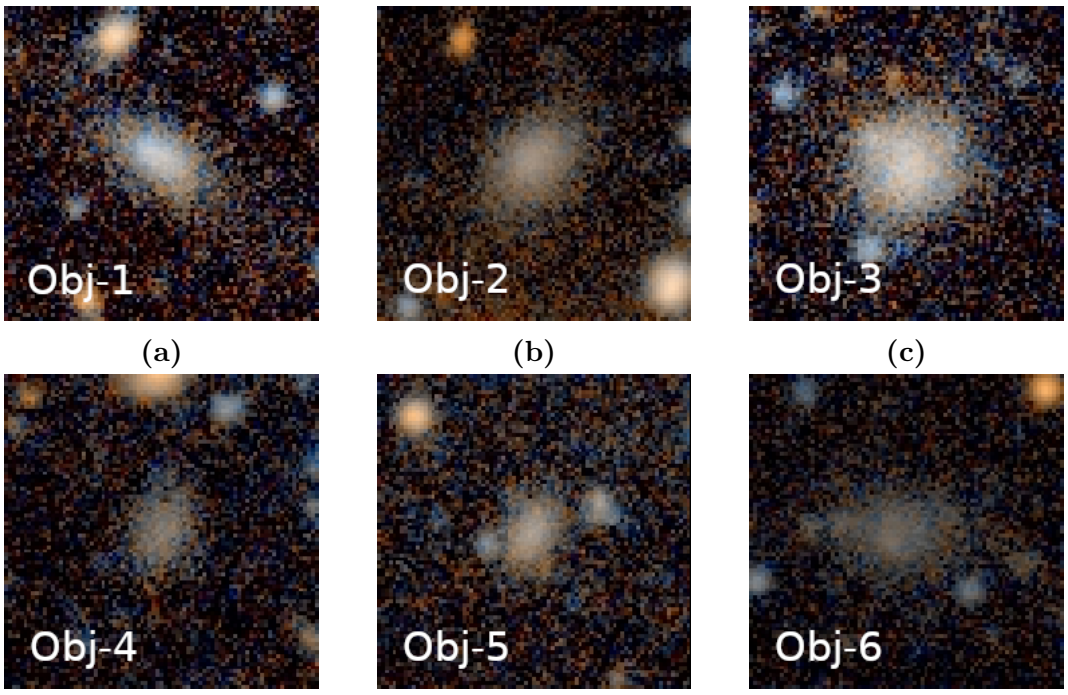


Fig. 4.1. Colored stamps of six Low Surface Brightness Satellites found in the selected region of the catalog. These images where made by making an intermediate 'green' filter consisting of the mean value of r and g filters and have a size of 18 arcsec.

As it was said, it can be finally confirmed that the algorithm works successfully with the results given. The ranges of effective surface brightness are within the ones of similar objects in the literature and the visual inspection test has been passed.

Table 4.1. Six examples of Low Surface Brightness Galaxies Satellites. This table shows the equatorial coordinates, effective surface brightness, surface brightness at half maximum and magnitude. These last three for g and r band.

Object	R.A (2000)	Dec (2000)	Eff SB (r)	Eff SB (g)	SB_{hm} (r)	SB_{hm} (g)	m_r	m_g
			$(mag/''^2)$	$(mag/''^2)$	$(mag/''^2)$	$(mag/''^2)$	(mag)	(mag)
Obj-1	02:40:28.4	-08:33:25.3	25.38	25.73	25.06	25.46	22.47	22.92
Obj-2	02:40:19.1	-08:32:48.2	25.46	26.05	25.15	25.76	22.11	22.86
Obj-3	02:39:37.7	-08:30:36.1	25.16	25.67	25.04	25.55	21.71	22.10
Obj-4	02:40:30.9	-08:29:50.7	25.96	26.62	25.81	26.49	23.37	23.89
Obj-5	02:40:27.6	-08:29:46.2	25.45	26.07	25.27	25.91	22.95	23.68
Obj-6	02:40:15.9	-08:18:43.2	25.89	26.51	25.72	26.25	23.07	23.47

Now that the final results have been presented, the focus can be shifted to future improvements of the code to detect more of these LSBGs and have a bigger sample. These improvements will be presented in the next chapter.

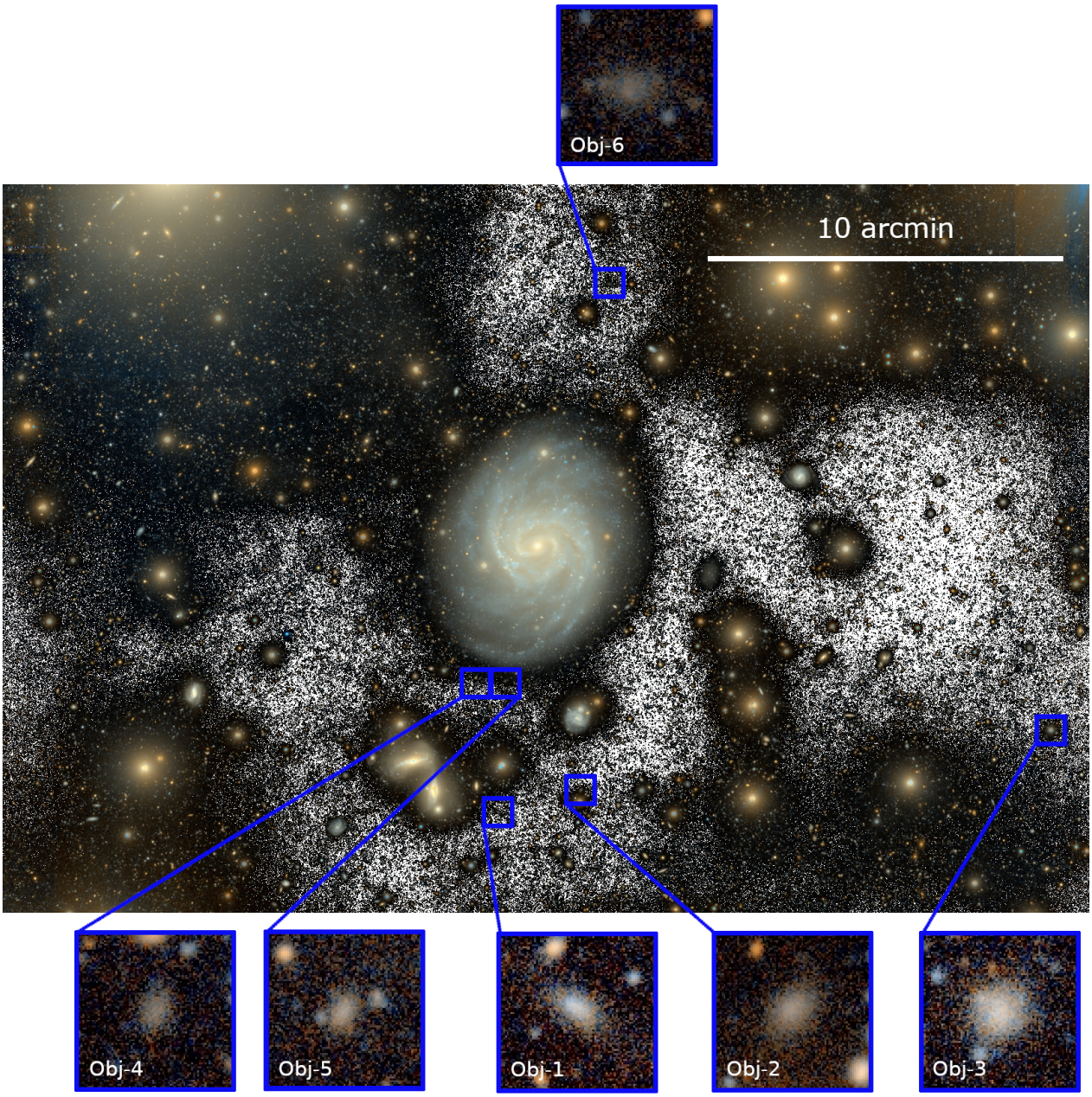


Fig. 4.2. Location of the six Low Surface Brightness Galaxy satellites around NGC1042.

Chapter 5

Conclusion

En el quinto capítulo del documento se analiza todo lo aprendido en el proceso de elaboración del trabajo, comenzando por una lista de todos los logros y habilidades desarrolladas en este. También se revisan los objetivos iniciales, lo cual lleva a plantear futuras mejoras para el algoritmo que se desarrollarán más allá del Trabajo de Fin de Grado.

Summarizing all the work developed during a project is a very good way of concluding and understanding which of the initial goals have been fulfilled. The next list collects all the achievements and the skills learnt in the process:

- Developing a Shell script pipeline capable of detecting Low Surface Brightness satellites. [1]
- Presenting a final sample of six low surface brightness galaxies **never observed before**.
- Learning **from zero** how to code in Linux shell and python, as well as automatizing shell scripts.
- Understanding the behavior of the Low Surface Brightness universe objects and the technical difficulties behind deep-imaging.

5.1 Objective evaluation

Regarding the objectives of the work, everything turned out very good. In the academic field all the objectives have been fulfilled. Developing an algorithm with a shell script pipeline has improved my perception of Linux and `gnuastro` tools, being a very helpful learning for future works. The algorithm itself has detected the objects which we were aiming for.

In the personal field, I have enjoyed every day in the process, working hand to hand with researchers and other students has helped me understand how this field works. As the project has not been finished yet and there are a lot of possible improvements to make, I will continue making progress and hopefully my results could be in a future paper.

[1] All the code scripts and pipeline used in this work can be publicly accessed in: https://gitlab.com/AndyGarciaSerra/trabajo-de-fin-de-grado

5.2 Future improvements

Writing a completely automatized algorithm that characterizes the objects of interest was an impossible task since the beginning. The algorithm was not meant to be automatic, having to stop in every step and double checking every parameter was well placed and the results were following a good process. This said, it is possible to improve these steps, making some changes in parts of the script that could automatize some of these tasks in a near future. In the next paragraphs these improvements will be explained.

First of all, a step forward in the algorithm will be implementing a section where color images are made with the stamps of the catalogue detections, like the ones we have shown at the end of the work. This will help making the identification process easier, as it is not only about the morphology of the objects but also about the shape and color. For this, an algorithm that detects the highest and lowest value of these stamps can be implemented to make better color images that make a better representation of the objects.

Another good improvement might be automatizing every detection in a pdf where all the parameters in the detection are gathered in an initial page. The next pages can collect all the objects detected and even classified by different measurements.

As some of the values in the measurements can be highly influenced by foreground and background objects, a good way of avoid this is, once the detections are located, to mask every sharp source around our objects and make a convolution of the resultant image, in which we can then make the measurements again.

Also, the catalogue making algorithm, which was developed in `Python`, can be easily implemented in the shell script pipeline, for a better and faster graph making. This code can even have different possible inputs and outputs so that every time a catalogue plot is needed anyone can easily make it.

Finally, a rejection algorithm for unwanted artifacts can be developed regarding the sharpness and another features. This will help clean future catalogues making all the process easier, as fortunately none of these artifacts will have to be inspected in the last steps of the process.

All these possible improvements will be taken in account in the near future to make a better, more automatized algorithm.

Bibliography

Akhlaghi, M. & Ichikawa, T. 2015, The Astrophysical Journal Suplement Series, 220, 1

Alam, S., Albareti, F. D., Allende Prieto, C., & Anders. 2015, The Astrophysical Journal Suplement Series, 219, 12

Bertin, E. 2006, Astronomical Data Analysis Software and Systems XV, 351, 112

Cohen, Y., van Dokkum, P., Danieli, S., Romanowsky, A. J., & Abraham, R. 2018, The Astrophysical Journal, 868, 96

Esposito, S., Riccardi, A., Fini, L., & Puglisi. 2010, SPIE, 7736, 773609

Esposito, S., Tozzi, A., Ferruzzi, D., et al. 2003, SPIE, 4839, 164

Martin, G., Kaviraj, S., Laigle, C., et al. 2019, Monthly Notices of the Royal Astronomical Society, 485, 796

Martin, N. F., Ibata, R. A., Lewis, G. F., & McConnachie. 2016, The Astrophysical Journal, 833, 167

McConnachie, W., A., Ibata, et al. 2018, The Astrophysical Journal, 868, 55

McConnachie, A. W. 2012, The Astrophysical Journal, 144, 4

Monelli, M. & Trujillo, I. 2019, The Astrophysical Journal, 880, L11

Tanoglidis, D., Drlica-Wagner, A., Wei, K., Li, T. S., & Sánchez. 2021, The Astrophysical Journal Supplement Series, 252, 18

Trujillo, I., D'Onofrio, M., Madrigal-Aguado, A., et al. 2021, Astronomy and Astrophysics, 14

Trujillo, I. & Fliri, J. 2016, The Astrophysical Journal, 823

van Dokkum, P., Danieli, S., Abraham, R., Conroy, C., & Romanowsky, A. J. 2019, The Astrophysical Journal, 874, L5

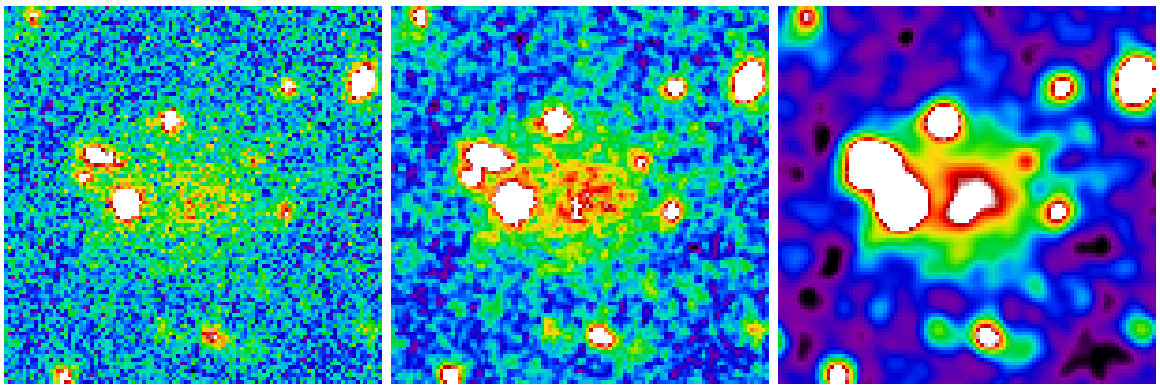
Appendix A

Detecting faint features using `Segment`

In this appendix, we will make an in depth description of how `gnuastro`'s `segment` works and how we can optimize it for the detection of our desired low surface brightness galaxies. To simplify things we will be using again the galaxy LBT1 (see Figure 2.4, last row) as an example of object we would like to be detected.

At first, we will focus on `segment` convolution step. This step takes the original image and smooths it given a convolution kernel, usually a Gaussian profile. These profiles can be created using `gnuastro`'s `MakeProfile`, which creates a mock 2D profile with the needed shape.

In Figure A.1, we have shown how the convolution step affects the original data. On the left image, we have the original data as a reference point before any convolution is applied. Taking a closer look at the second image, which is the original data after the default convolution, we can clearly see how this sharp convolution kernel (1.5 pixels FWHM gaussian) has affected mostly the point like or very sharp sources of the image, smoothing the contrast of these acute peaks. The opposite happens when applying a wide convolution kernel, as we did in the last image, where we can see the influence of the custom kernel (5 pixels FWHM gaussian) at the diffuse part, making the center part an smoother surface.



(a) **(b)** **(c)**

Fig. A.1. Examples for `segment` convolution step. In **(a)** we have the original image data, **(b)** is the default convolution of `segment` (Gaussian of 1.5 pixels FWHM) and **(c)** is the image after a custom kernel convolution (Gaussian of 5 pixels FWHM). All three images where exported from `ds9` using `sls` color map and `zscale` contrast scale.

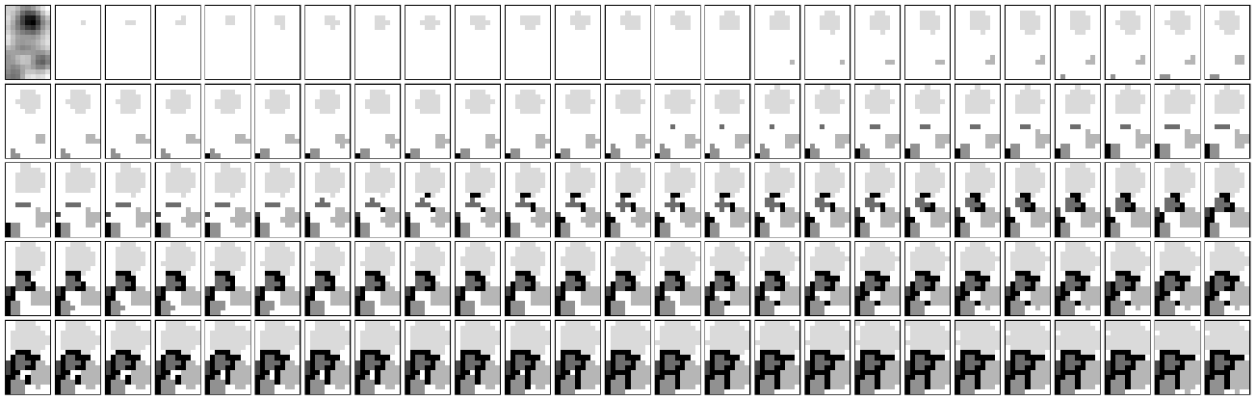


Fig. A.2. Representation of how `gnuastro`'s `segment` program makes a pixel-by-pixel clump differentiation with an input convoluted image. Image taken from Akhlaghi & Ichikawa (2015).

This custom kernel differences are the key for `segment` to detect the diffuse part of the image as we need and this happens because of how `segment` works behind the scenes taking this convoluted image and turning it into a segmented map of labeled clumps. Now that we have illustrated the importance of finding the right kernel for the convolution step we can focus on the influence of this kernel selection in the clumps-making step of `segment`.

In this step, `segment` splits up data in different tiles and, starting by the brightest pixel in this tile, starts labelling, going from the brightest to the faintest pixel of the tile. If two of these pixels are side-by-side, the label is still the same, but when a bright pixel appears (as another peak in the tile) it gets a different label. Once these labels are established and the distinct labels touch each other, segment draws a frontier between these so all the different labeled clumps are clearly differentiated. This process is illustrated in Figure A.2, where clumps are made from a real data image. In Akhlaghi & Ichikawa (2015) there is an accurate and more extended description of the performance of `NoiseChisel` and `segment` on the detection of nebulous objects.

Having this process in mind, we can figure out that, whenever we fail in making the correct convolution kernel for our needs this clump-making process can split what we need to be in one clump into different clumps. We can now jump back to Figure A.1, where we can see the default convoluted image in the middle. This image has a deficient kernel making the central part not uniform, so that `segment` will split this central part in multiple clumps. This will not happen in the convoluted image we showed in the right, where the convolution kernel was wider, making a uniform central part that will fit into a single clump, as we need.

To clarify this last argument, we can jump back to in Figure 3.4 (section 3) were we represent the clump detection for each convolution. In the second column of the image we can see how `segment`'s behavior changes with different convoluted images. With the default convolution, which corresponds to the second row, we can see how the diffuse part in the center gets detected as a huge number of little clumps whereas with the custom kernel convoluted image shown in the first row the central clump is much bigger because of the uniformity achieved after smoothing with a wider kernel we were talking about earlier.

This leads us with the third column of Figure 3.4, in which the final clumps are shown. These clumps are the ones that would make it to the catalog of objects once we finish with

`segment`. As we can see with the default convolution, the central diffuse region is not making it to the final clumps image, while with the custom kernel convolution it clearly is.

The intermediate step that discards the smaller clumps is done by making the same clump detection as we have shown but in the undetected region of the image instead of the detected region. In these region full of noise the clumps will have much peaks and different random light shapes that generally will develop into a huge number of small clumps. Then the algorithm compares the clumps of the detection region with the clumps of the noise region, discarding all the clumps in the detection map that are similar size as the ones in the noise region, so that the noise inside any signal region is discarded (in Figure A.3 there's an example of how noise clumps look like).

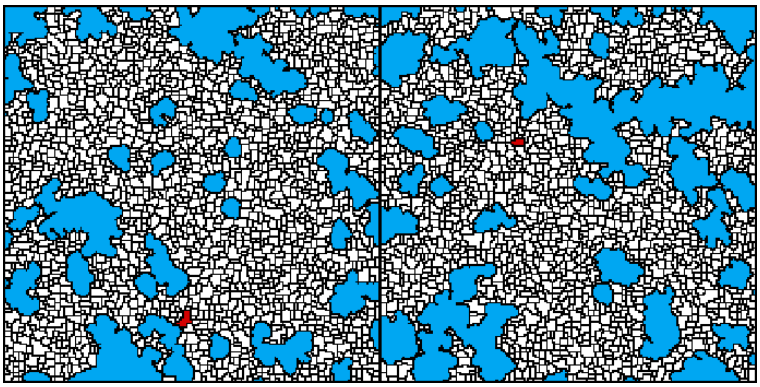


Fig. A.3. Simple representation crop of how `segment` splits noise in clumps to discard noise inside the detection regions. In blue we can see the detections region and in white we can see all the clumps related to the noise regions.

That is why a nicely configured custom convolution kernel is very important for any case. If we do not convolve the diffuse regions with a widely enough kernel, the part we are interested in will just be discarded because of how similar it will be to the noise in the eyes of the algorithm. We can say that the segment convolution step is the key step in all the detection process because is the one that shapes your detections properly making the catalogues you are looking for.

Appendix B

Sky determination using `NoiseChisel`

In this Appendix we will explain in detail how `gnuastro NoiseChisel` tool separates signal from noise. This will give us a better understanding on the importance of a good chiseling step when trying to detect faint structures.

First of all we can take a closer look at the top left corner of the weighted image we were presenting back in chapter 3, Figure 3.2. For this galaxy, which is NGC1052, we can clearly see a halo spreading on its lower right side. We will focus in this halo as an extreme example so we can understand the importance of the behind the scenes steps `NoiseChisel` makes to differentiate signal from noise.

What `NoiseChisel` does as a first step is separating all the input image pixels into tiles of 30 by 30 pixels, in the default settings, and comparing these tiles with the ones around. These tiles surrounding the tiles of interest are called neighbours and the number of neighbours with which `NoiseChisel` compares this central tile is 15 as default.

When making this comparison `NoiseChisel` takes in account the difference in the quantile of the mean and the median values of the pixels inside the tile. These quantiles are nothing more than the proportion of the total data on each side of the mean and median for each distribution. This means, the median will always be at the center of the distribution, having his quantile 0.5, but the mean value does not have to be in the center.

In a region ruled by noise, the distribution of the pixels is going to be similar than a Gaussian, with a clear maximum and a more-or-less symmetrical profile in both sides of the median. However, when signal is added to this region, the distribution is shifted to higher pixel values, as the signal is going to be brighter than the noise. This ends in a higher mean quantile that differentiates from the image without signal. These distributions are shown in Figure B.1.

In this Figure, we can see three different examples of mock galaxies where they plotted the distribution for the signal+noise image and the signal only image. We can see how this signal adding is shifting the distribution to the right. Last galaxy is the best example to see this. This has a high effect on the value of the mean.

For instance think about a simpler distribution of numbers, like ones and zeroes. If you add a number 10 to this distribution, the mean value of it will rapidly shift to a higher value, whereas the median will not be that influenced. This principle is what `NoiseChisel` takes

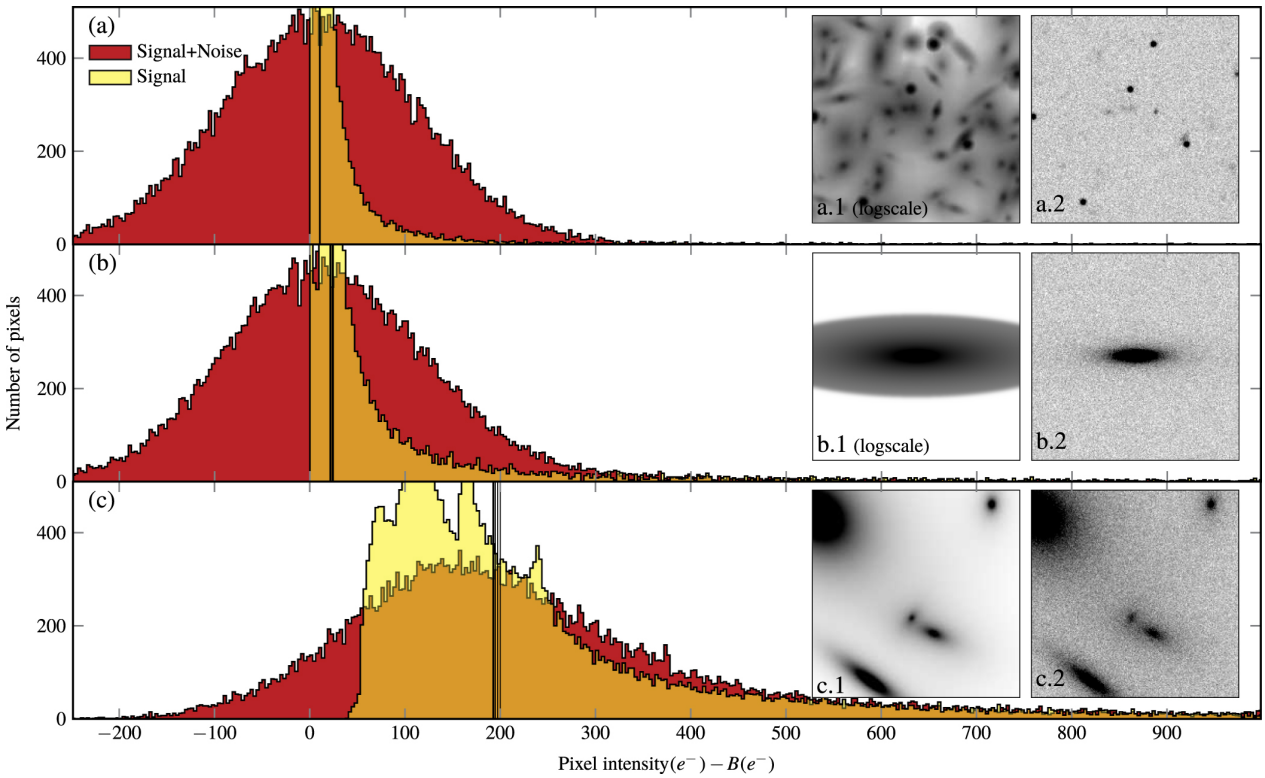


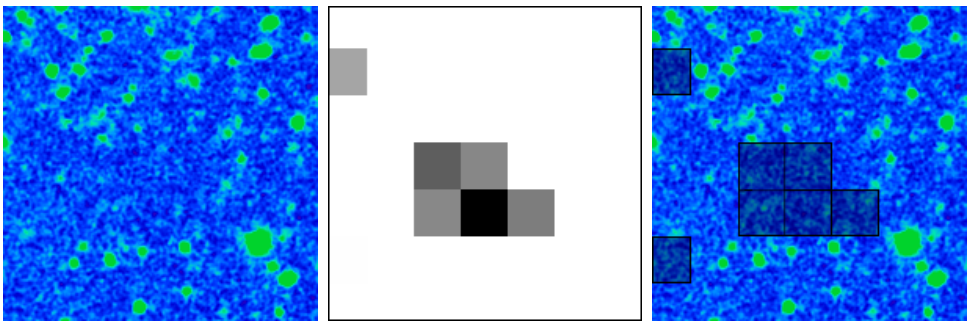
Fig. B.1. Different mock images with their signal and noise pixel distribution. Image taken from Akhlaghi & Ichikawa (2015).

advantage of to make a differentiation between signal and noise.

First of all `NoiseChisel` takes a look at the pixels inside each tile and discards the tiles which have relatively different mean and median quantile values, because those will correspond to signal, leaving the tiles with close values of mean and median quantiles, which are tiles potentially containing noise. The default parameters of the algorithm allow a differentiation of 0.01 between both quantiles, letting the quantile of the mean oscillate between 0.51 and 0.49. It is important to make the point that by definition the quantile of the median is always 0.5.

In Figure B.2 we can see how these tiles are made following a real signal image cropped from the weighted one we showed in Figure 3.2. This crop was made on the result of applying `NoiseChisel` default parameters on the original data and it's easy to see that any tile containing a sharp source or any kind of signal is rejected, leaving just the tiles without them. This confirms that `NoiseChisel` default parameters work well for this particular region of the image.

The real handicap comes when we are interested in a very homogeneous signal region like the wing of a galaxy or a galactic halo. When this happens, all the signal has a similar distribution as the noise we showed before, so even if they are not sharp peaks on the region, the image holds important data, that will be rejected if we stay with the same default parameters. To fix this problem we can change the parameter that allows a wider range of values for the mean quantile, which is called `--meanmedeqdiff`, but it woll not really affect the results when we are working with such large flat surfaces.



(a) **(b)** **(c)**

Fig. B.2. Tile assignation examples with a real signal image from the LBT NGC1042 data. (a) Represents the signal stamp, (b) the tiles assigned as noise and (c) is a composition of both images for a better visual inspection.

What we need to do is changing the size of the tiles that the algorithm uses to make this comparison, so that this size is similar or at least has the same order of magnitude as the objects we are interested in. This last parameter, called `--tilesize`, will help us detect the regions we are really interested in.

Another important parameter to have in mind is `--interpnumngb`, which is the one that the algorithm uses after the tiles are kept, the next step after this will be a comparing step of the tiles kept and the tiles surrounding them. This parameter tells the alrogithm the number of neighbours to compare with.

These three parameters are the key for a good signal and noise segregation, which is very important if you do not want any of your data to be treated as noise and loose information in the detection process.