PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Using clustering for disperse objects fields segmentation in MIRADAS instrument

Josep Sabater, Santiago Torres, Francisco Garzón, José María Gómez

Josep Sabater, Santiago Torres, Francisco Garzón, José María Gómez, "Using clustering for disperse objects fields segmentation in MIRADAS instrument," Proc. SPIE 10707, Software and Cyberinfrastructure for Astronomy V, 107070N (6 July 2018); doi: 10.1117/12.2312528

Event: SPIE Astronomical Telescopes + Instrumentation, 2018, Austin, Texas, United States

SPIE.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/27/2018 Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

Using clustering for disperse objects fields segmentation in MIRADAS instrument

Josep Sabater[a,b], Santiago Torres[b], Francisco Garzón[c,d], and José María Gómez[a]

[a]Institut de Ciències del Cosmos, Univ. de Barcelona (IEEC-UB), Barcelona, Spain
[b]Depto. Ing. Informática y Sistemas, Univ. de La Laguna, La Laguna, S.C. Tenerife, Spain
[c]Instituto de Astrofísica de Canarias, La Laguna, S.C. Tenerife, Spain
[d]Departamento de Astrofísica, Univ. de La Laguna, S.C. Tenerife, Spain

ABSTRACT

Mid-resolution InfRAreD Astronomical Spectrograph (MIRADAS) is a near-infrared multi-object spectrograph for Gran Telescopio Canarias (GTC). It has 12 deployable Integral Field Units (IFU) based on probe arms with pick-off mirrors, each of which can observe a different user-defined sky object. MIRADAS can work with target sets where their components are spread over such a wide area so that all of them do not fit in the field-of-view of the instrument. Therefore, data sets of that kind require, prior to capturing them, some arrangement that groups its elements in different subsets where the distance between the two most remote elements is inferior to the field-of-view diameter. This field segmentation is achieved using a hierarchical clustering technique. Our method relies on determining mutual nearest-neighbors, which will be merged if they show a given degree of similarity known beforehand. Moreover, we also compute a geometric center for these clusters, information to be delivered to the telescope's pointing process. This step is formulated as the minimum bounding disk problem, which founds the center of the smallest radius circle enclosing all points of a cluster. Finally, we consider several real science cases and analyze the performance of the proposed solution.

Keywords: disperse objects fields, cluster analysis, telescope pointing, probe arm, multiple-object spectrography

1. INTRODUCTION

Many of the leading scientific open questions in modern Astrophysics can be only answered with the help of moderate to high spectral resolution instruments mounted in large collecting area telescopes. As a result, the demand for instruments with multi-object spectrography (MOS) capabilities has significantly increased in the last decades. The Mid-resolution InfRAreD Astronomical Spectrograph (MIRADAS) is a near-infrared MOS for Gran Telescopio Canarias (GTC).[1] This instrument is under development by an international consortium led by University of Florida and is provisionally scheduled for commissioning at the end of 2018. One of its more remarkable components is its multiplexing system (MXS), which enables it to simultaneously observe up to 12 user-defined astronomical targets. This task is accomplished by 12 independent and deployable Integral Field Units (IFU) based on robotic probe arms with pick-off mirrors working in a cryostat[2] (see Figure 1). Each of these arms has a series of folds inside the articulated mechanism that relay light from the telescope focal plane to the spectrograph. These internal optical elements are designed for simplicity, always keeping a fixed optical path length while the arm is moving between any two locations. Furthermore, the tubes and bars forming the arm are arranged in a way that provides a high level of stability to the mechanical structure when it works upside down. However, the kinematic behavior of the arm is not intuitive, making the control of the optomechanical system more complex.[3]

To reach the targets, all arms in the system need to execute safe trajectories starting from their current positions. As long as all targets can be fitted inside the region of the MIRADAS field-of-view (FOV), the arms motions can be computed following a two-step sequential process. First, a piece of software known as target allocator analyzes each of the targets and delivers a sequence of assignments, a set of <*target, arm*> pairs.[4] The

Send correspondence to Josep Sabater - jsabaterm@el.ub.edu

Software and Cyberinfrastructure for Astronomy V, edited by Juan C. Guzman, Jorge Ibsen, Proc. of SPIE Vol. 10707, 107070N · © 2018 SPIE · CCC code: 0277-786X/18/$18 · doi: 10.1117/12.2312528

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/27/2018
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

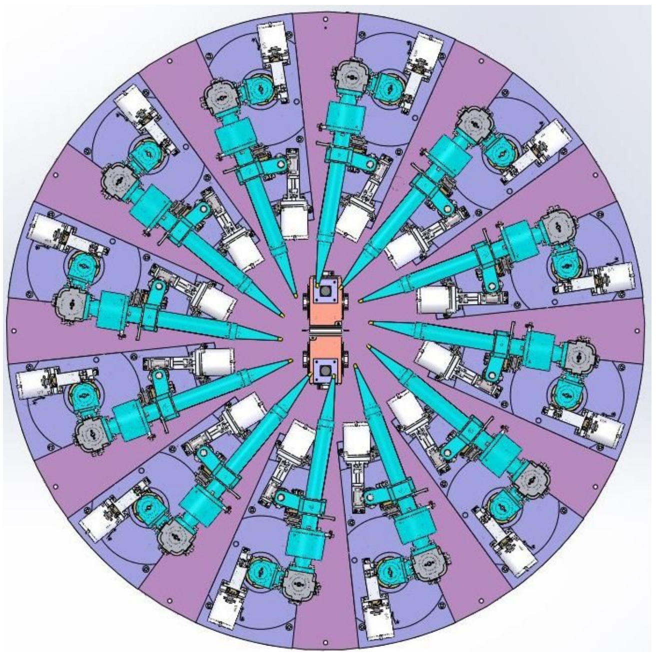


Figure 1: A drawing of the MIRADAS MXS bench with 12 probe arms. The primary task of these mechanisms is to relay light from a given target located in the instrument's FOV to the spectrograph. The FOV is a circular area with a diameter of 250mm placed in the middle of the bench.

number of pairings returned mainly depends on the location of the targets as well as the kinematic characteristics of the arms, but they cannot be greater than the number of arms available. Then, the motion planner determines collision-free trajectories for each of the arms in the assignment plan. Multiple iterations of this two-step process might be required, especially if there are more targets than arms in the system. However, since MIRADAS does not severely constrain the star fields it can work with, the user-defined targets might be spread over an area more extensive than the instrument's FOV. In such cases, an additional processing step has to be carried out before the two previously mentioned. This step is responsible for finding some sort of arrangement that properly distributes the targets in different groups so that all elements in a batch fit in the FOV region. Hence, it makes sense that this preprocessing task groups targets following a geometrical approach: the distance between any pair of elements in a set has to be less than or equal to the FOV diameter. In addition, for each batch, it has also to be computed practical information that will help the telescope's pointing process to adequately place the MIRADAS FOV center so that all targets of the batch are reachable.

The partition of a data set of interest into distinct subsets so that members within a group are related to one another is known as cluster analysis,[5] a well-established technique in unsupervised machine learning and data mining. There exist a number of methods; however, distance-based algorithms are very frequent in real applications due to their simplicity and ease of implementation. K-means[6] is one of the most broadly applied algorithms in this family, but presents a few limitations. One of them is that the algorithm requires the specification beforehand of the number of distinct clusters to extract from the input data set, which is what we precisely want to determine in MIRADAS. Hierarchical agglomerative clustering algorithms overcome this limitation. They produce a hierarchic relationship of the points to cluster, requiring the execution of an additional method to select the optimal number of clusters. This hierarchy is frequently represented utilizing a *dendrogram*.[*] After visually or analytically analyzing it, data scientists cut the tree at a given height/distance, yielding a number of clusters. Other times, once computed the complete dendrogram, the user decides the number

[*]A tree specifying the relationships between the data in the set. It has many leaves at one end corresponding to all points in the input data set. As we move toward the other end of the tree, similar data is progressively merged into branches until reaching the root, which contains all points. Sometimes, for convenience, the tree is displayed horizontally.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/27/2018
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

of clusters and the tree is cut accordingly. However, in the case of MIRADAS, there is no need to construct the full tree as we know well in advance the height where it has to be cut. Unfortunately, to our best knowledge, popular cluster analysis software packages/libraries do not allow the direct construction of a height-limited tree.

In this paper, we describe the particular threshold-assisted agglomerative algorithm used in MIRADAS to automatically determine the distinct batches of targets. It is based on the search of *Reciprocal Nearest Neighbors* (RNN) by constructing chains of Nearest Neighbors (NN-chain). Introduced in the early 1980's,[7,8] this technique guarantees $O(n^2)$ worst case time complexity[†]. However, knowing the celestial targets forming each cluster is not enough. As we have commented before, to fully characterize each group, in addition to the elements it comprises, its center needs to be also computed. Cluster algorithms sometimes do not provide this information. In other occasions, a center can be statistically inferred from the clustering process, but it could not meet the geometrical constraints of our problem. Here we also present the approach we have followed to compute the batches' centers, which relies on an algorithm determining the smallest circle containing all the elements in a cluster. Bounding volumes are employed in many fields, especially in ray tracing and motion planning. In the latter, these simpler shapes are used to speed up collision checking between a robot and the environment or among two or more robots.

The paper is organized as follows. First, we introduce hierarchical agglomerative clustering as well as the enhancement applied to detect the number of clusters automatically. Next, we provide the approach followed to determine the center of each cluster. In Section 4, we demonstrate the performance of our approach. Finally, we conclude with future work.

2. AGGLOMERATIVE CLUSTERING WITH NN-CHAINS

Although different agglomerative methods can be found in literature, all employ a *similarity* matrix. Additionally, all follow a common bottom-up *greedy* strategy that starts by separating each component of the input data set into its own individual cluster. Then, at successive iterations, the two most similar clusters are combined into a new one. This sequential process ends when there is only one cluster left comprising the entire initial data set. The approaches only differ in two points: (i) the measure used to compute the *similarity* between two elements in the data set and (ii) the inter-cluster *linkage criteria*, which specifies how distance between each pair of clusters is determined. There are many similarity metrics[10] as well as linkage schemes.[11] Although intuitive, this greedy approach presents two computational problems. It requires a significant amount of storage as it works with a $O(n^2)$ similarity matrix and has a time complexity of $O(n^3)$, where n is the number of points in the data set.

Popular software packages such as *scikit-learn* or *scipy* in Python and *cluster* in R implement agglomerative clustering. However, to our knowledge, those pieces of software either (a) return the complete *dendrogram*, and it is the analyst who decides the height of the cut, yielding, this way, a number of different clusters or (b) the number of clusters to find is specified as an input parameter to the clustering routines. Considering the geometrical constraints in MIRADAS, a building process that automatically prevents the agglomeration of two clusters if their dissimilarity is larger than a given value (the diameter of the instrument FOV) seems more appropriate. Such an approach is frequently utilized in image analysis and object recognizition.[12,13] This hierarchical agglomerative scheme including a cut-off threshold is based on the detection and agglomeration of RNNs,[7,8] which are pairs of points i and j such that i is j's nearest neighbor and j is i's nearest neighbor. The search process is carried out with the help of a structure know as NN-chain. It is a sequence of clusters composed of a random agglomerable cluster followed by its NN, which is then followed by its NN from the remaining agglomerable clusters, and so on.

The NN-chain approach merges clusters in a different order to the one employed in the common greedy strategy described in a previous paragraph. However, for those linkage criteria fulfilling the Bruynooghe's reducibility property,[10] the NN-chain approach is the preferred since it achieves the same results in $O(n^2)$ time.[14,15] We adopt the Euclidean distance and pursue *complete-linkage*, which meets the reducibility property. In this connectivity scheme, the distance between two clusters is equal to the furthest distance between a member of one cluster and a member of the other. Pseudocode for the threshold-based agglomerative clustering approach

[†]In computer science, the time complexity is the computational complexity estimating the running time of an algorithm. It is expressed using *big O notation*, where n normally refers to the input size.[9]

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/27/2018
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

is given in Algorithm 1. As can be appreciated, the algorithm arbitrarily chooses an initial cluster (line 2) and then, with the help of a routine that finds nearest neighbors (line 5), iteratively builds up an NN-chain (line 7). Once a RNN pair is discovered (line 8), the involved clusters are agglomerated if they show a similarity compatible with a given threshold (line 10). Otherwise, the chain construction is interrupted and all its elements are included in the final cluster sequence (line 13). What really makes this implementation more efficient than the standard greedy approach is that the NN-chain can be reused in successive iterations. That is due to the reducibility property, which ensures that the merging of a given RNN pair does not affect the relations among the NN discovered before. Note that the pseudocode provided does not consider what has to be done if a premature end occurs. In fact, it may be possible that the while loop (line 4) exits when the current NN-chain is not empty. In such case, the remaining clusters need to be inspected again to check if they can be agglomerated further.

Algorithm 1 Threshold-based agglomerative clustering using NN-chains

```
Require:
    $\mathcal{P}$ : a sequence with all points in the data set, where $p \in \mathcal{P} = (p_1, \ldots, p_n)$
    $t$: agglomeration threshold
Ensure:
    $\mathcal{C}$: a sequence with the clusters found

 1: $\mathcal{C} \leftarrow \varnothing; last \leftarrow 0; lastSim[0] \leftarrow 0$
 2: $\mathcal{N}[last] \leftarrow p \in \mathcal{P}$                    ▷ Initialize NN-chain with an arbitrary point
 3: $\mathcal{R} \leftarrow \mathcal{P} \backslash p$                 ▷ Sequence with the remaining agglomerable clusters
 4: while $\mathcal{R} \neq \varnothing$ do
 5:     $(c, sim) \leftarrow$ getNearestNeighbor($\mathcal{N}[last]$, $\mathcal{R}$)
 6:     if $sim > lastSim[last]$ then                ▷ Update NN-chain with the cluster
 7:         $last \leftarrow last + 1; \mathcal{N}[last] \leftarrow c; \mathcal{R} \leftarrow \mathcal{R} \backslash c$
 8:     else                                          ▷ RNN pair found
 9:         if $lastSim[last] > t$ then
10:             $c \leftarrow$ merge($\mathcal{N}[last]$, $\mathcal{N}[last - 1]$)
11:             $\mathcal{R} \leftarrow \mathcal{R} \cup c$; $last \leftarrow last - 2$
12:         else                   ▷ Update cluster sequence with the content in NN-chain
13:             $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{N}$; $last \leftarrow -1$; $\mathcal{N} \leftarrow \varnothing$
14:         end if
15:         if $last < 0$ then          ▷ Initialize a new NN-chain with a random cluster
16:             $last \leftarrow 0$; $\mathcal{N}[last] \leftarrow p \in \mathcal{R}$; $\mathcal{R} \leftarrow \mathcal{R} \backslash p$
17:         end if
18:     end if
19: end while
20: return $\mathcal{N}$
```

3. SMALLEST BOUNDING CIRCLE FOR A SET OF POINTS

Determining the circle of smallest radius enclosing a set of points is a classical optimization problem of computational geometry. As a circle is uniquely determined by at most three points, a brute force approach can be attempted. It will consist in checking every circle defined by three and two points and keeping the minimum-sized disk containing all points of the set. This is by far the worst solution since there exist $O(n^3)$ circles and $O(n)$ time is required to check each of them, yielding, then, a total execution complexity of $O(n^4)$. Although alternative methods can be found in literature, it was Welzl[16] the first proposing a fast and easily implementable algorithm for 2D. His solution was inspired by a previous work solving a linear program with n constraints and d variables in *expected* $O(n)$ time, provided d is constant. In such work as well as in Welzl's work a randomized incremental construction is used, a powerful technique nowadays widely employed to efficiently solve optimization problems

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/27/2018
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

found in computational geometry. Although it provides a convenient framework, it comes at the cost that the $O(n)$ running time is only an *expected* bound that might be larger for some data sets. Generally, this technique applies to those optimization problems where: (i) the solution does not change if a new constraint is added or (ii) the solution is partially defined by the new constraint so that the dimension of the problem is reduced.[17] Both conditions are met in the smallest enclosing disk problem.

Given a set of n points $P_n = \{p_1, p_2, \ldots, p_n\}$, the Welzl approach iteratively selects one point and grows the smallest bounding disk that contains that point and the previously processed. The algorithm is based on the observation that when the smallest disk D_i enclosing the subset $P_i = \{p_1, p_2, \ldots, p_i\}$, where $1 \leq i \leq n$, is updated to include another point p_{i+1}, then the new optimal disk $D_{i+1} = D_i$ if the point belongs to D_i. Otherwise, D_{i+1} must pass through p_{i+1}. In the former case, the algorithm directly proceeds to the next point while in the latter a new method computing the smallest bounding circle of a set of points P with a given point p_i on its boundary is required. Relying on this additional method that we will call `minidiskWithSupport(`P, p_i`)`, Welzl incrementally computes the optimal disk for the initial data set.

The algorithm for `minidiskWithSupport(`P`, `p_i`)` follows a similar iterative approach. It sequentially checks the points of the set for inclusion, calling, when necessary, a subroutine determining the smallest enclosing circle of a set of points with two given points in its boundary. This new subroutine is also constructed using the same framework, but this time it will require another subroutine computing a bounding disk of a set of points with 3 points in its perimeter. As a circle is determined by at most three points, there is no need to expand further the search. Although this scheme of nesting routines is quite intuitive, a more compact recursive solution can be obtained relying on a generalized version of `minidiskWithSupport(`P`, `R`)`. This version determines the smallest disk containing the points in P with the points in R on its perimeter (see Ref. 16 for proofs). The pseudocode for the recursive approach is shown in Algorithm 2. The smallest enclosing disk for a set of points P can be solved by the call `minidiskWithSupport(`P`, `$\varnothing$`)`.

Algorithm 2 Smallest enclosing disk

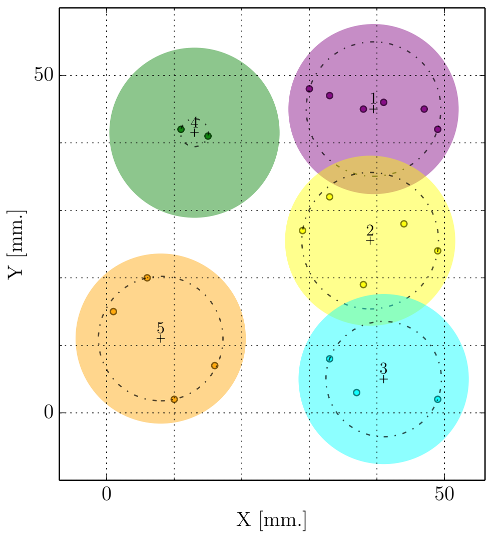
```
Require:
    $\mathcal{P}$ : a set with all points to be enclosed
    $\mathcal{R}$: a set with all points that must be on the boundary of the smallest circle
Ensure:
    $\mathcal{D}$: The smallest circle enclosing the set $\mathcal{P}$

 1: function MINIDISKWITHSUPPORT($\mathcal{P}$, $\mathcal{R}$)
 2:     if $\mathcal{P} \neq \varnothing$ or $|\mathcal{R}| = 3$ then
 3:         $\mathcal{D} \leftarrow$ BUILDCIRCLE($\mathcal{R}$)          ▷ Construct a circle with the given points in its boundary
 4:     else
 5:         select $p \in \mathcal{P}$ randomly
 6:         $\mathcal{D} \leftarrow$ MINIDISKWITHSUPPORT($\mathcal{P} \backslash p$, $\mathcal{R}$)
 7:         if $p \notin \mathcal{D}$ then                                 ▷ point $p$ does not lie inside $\mathcal{D}$
 8:             $\mathcal{D} \leftarrow$ MINIDISKWITHSUPPORT($\mathcal{P} \backslash p$, $\mathcal{R} \cup p$)
 9:         end if
10:     end if
11:     return $\mathcal{D}$
12: end function
```

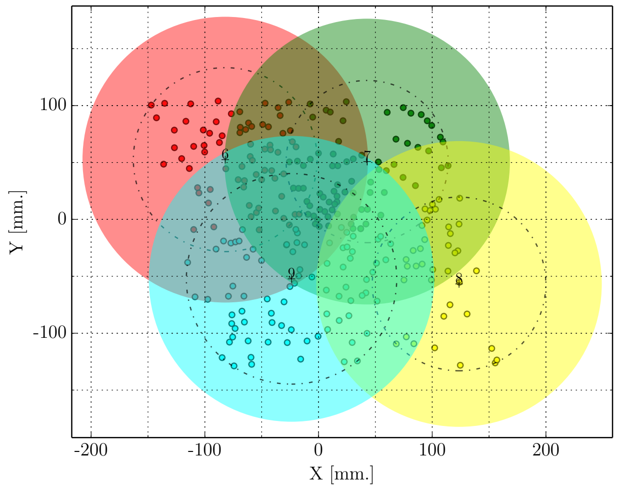
4. SIMULATIONS AND RESULTS

The proposed threshold-based RNN clustering algorithm for *complete-linkage* as well as the minimum bounding disk, respectively described in Section 2 and in Section 3, were implemented. Several simulations with synthetic and with real science targets were conducted to study the performance of our method. First of all, we will focus

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/27/2018
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use



(a) Five different clusters, each comprising the small dots of the same color. For each of the clusters, we show its center (black cross), the smallest enclosing disks (dashed circle) used to compute the center as well as a larger colored circle centered at the corresponding cross with a radius equal to *threshold/2* (12.5mm). The numbers above the cluster centers specify the cluster name.



(b) Four overlapping clusters were found for this scenario with 270 targets. Many targets that were initially assigned to different clusters (distinct colors) are inside more than one *threshold* circle (in bold color).

Figure 2: Synthetic data scenarios.

on two random scenarios, which will help us to show the reader what we want to accomplish in MIRADAS by employing the proposed algorithms. In Section 4.2, our solution is tested with real celestial objects.

4.1 Synthetic scenarios

In the first of the two synthetic scenarios, a total of 20 points were arbitrarily arranged inside a square of side 50mm, as seen in Figure 2a. Then, the clustering algorithm with a threshold value of 25mm. (the FOV diameter for this example[‡]) is run to detect clusters automatically. Once known these clusters and the points included in each of them, the smallest enclosing circle is determined for every cluster. The center of this circle will be used as telescope's pointing information while the members of each cluster will be delivered to a different instance of the *target allocator* since it is assured that all targets in a cluster can be fitted in the FOV area. Intra-cluster and inter-cluster distances for this example are given in Table 1. The former refers to distances between elements in a cluster, while the latter refers to the distance between two clusters, which is the distance between their two farthest-apart members. As expected, the dissimilarity (distance) between any pair of members of one cluster is inferior to the specified threshold while the dissimilarity between clusters remains higher than the threshold. In addition, the results show how the algorithm generates compact groups. That is due to the tendency of the selected linkage scheme to combine those cluster pairs whose merge has the smallest diameter. Finally, we plot the full dendrogram, given in Figure 3, for this scenario. As seen there, if the tree is properly cut at a height equal to 25, we obtain the 5 clusters found in Figure 2a.

A second random scenario, but this time with 270 targets more densely distributed was generated. Clustering with a cut-off threshold of 250mm was performed. As can be appreciated in Figure 2b, four clusters were needed to contain all points in the input data set. For some clusters, their corresponding circles representing the FOV

[‡]The worst case scenario for the presented agglomerative algorithm is that presenting three points separated all of them by a distance equal to the FOV diameter. To guarantee, even in such case, that all points in a cluster are contained in a given FOV, the cut-off threshold should be equal to the side of the equilateral triangle inscribed in a circle with diameter equal to the FOV diameter.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/27/2018
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

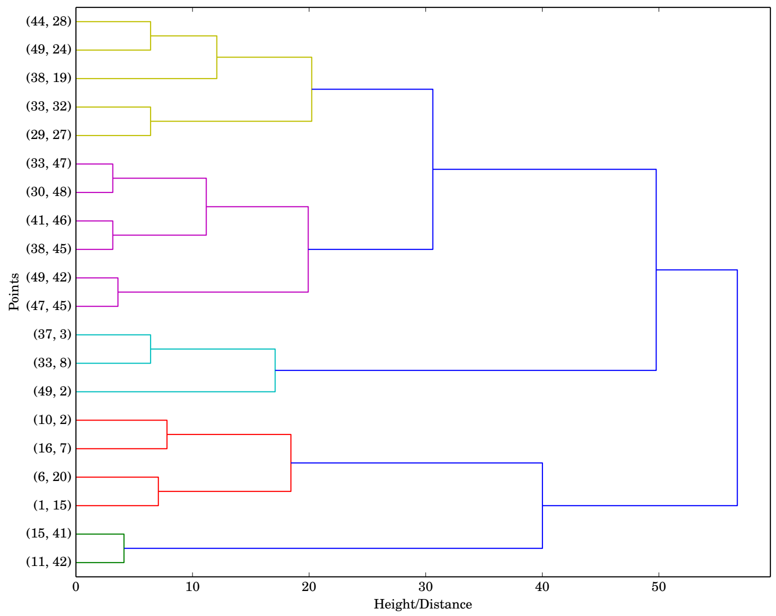


Figure 3: Dendrogram for the clustering example in Figure 2a. The tree has as many leaves as points are in the scenario and they are merged up to the root, where all the initial points are members of a single cluster. A cut at a distance of 25 gives the 5 clusters automatically obtained by our algorithm.

overlap. That means that targets in the intersection area of two or more FOV, although belonging to different clusters, can be delivered to two or more distinct instances of the target allocator. This degree of flexibility might be conveniently exploited when trying to compute safe trajectories for the targets. In fact, as we will see, this superposition is normal when clustering real science targets since they are frequently arranged in a quite compact surface.

4.2 Real data scenarios

We have also performed a series of tests with real data. They contain groups of celestial objects that, according to the MIRADAS science cases, can be targeted by the instrument. These sky targets were originally determined in equatorial coordinates. To be represented in the surface of the MIRADAS FOV, they were previously converted into standard coordinates (aka tangential coordinates) using the MIRADAS plate scale. Since each input data set does not implicitly specify the center/reference of the targets to observe, we considered it to be the mean value of the samples. Once known the ξ and η values, the X and Y axis respectively, the data was delivered to the clustering algorithm. In Figure 4a, the cluster analysis for several objects belonging to Messier 13 (M13) is given. All targets fit in the MIRADAS FOV area; therefore, a single cluster, with its center at the denser zone of the data set, was found.

A different scenario is shown in Figure 4b. This time targets represents a group of objects in Czernik 3. The clustering algorithm returns more than one cluster since the targets occupy an area slightly bigger than the MIRADAS FOV. Note that these clusters present bias. That means that the members of a cluster can be

Table 1: *Intra-cluster* and *inter-cluster* distances for the example in Figure 2a

(a)

	Intra-Cluster dist. (mm.)	
Cluster	min.	max.
1	3.1622	19.9249
2	6.4031	20.2237
3	6.4101	17.088
4	4.1231	4.1231
5	7.8102	18.4391

(b)

	Inter-Cluster distance (mm.)				
	cluster 1	cluster 2	cluster 3	cluster 4	col5
cluster 1	...	30.6105	49.7695	30.2655	36.8782
cluster 2	30.6105	...	34	42.0476	29.5466
cluster 3	49.7695	34	...	55.1725	46.6154
cluster 4	30.2655	42.0476	55.1725	...	28.7924
cluster 5	36.8782	29.5466	46.6154	28.7924	...

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/27/2018
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

concentrated in a peripheral zone of the FOV. That is the case of the red cluster, where its components are mainly concentrated in the center and towards the inferior semicircle of the FOV. This bias, if necessary, could be mitigated by adequately rebalancing the load of those clusters presenting overlapping areas. The effect of bias can be more marked, especially in scenarios with a larger number of targets. Figure 4c shows 1354 objects from Sagittarius A (Sgr A), all densely distributed in a surface moderately bigger than the MIRADAS FOV. The corresponding circles associated with each cluster are given in Figure 4d and bias is more significant here than in the previously analyzed cases.

Finally, we retrieved two more sparsely distributed sets of sky objects from the UKIRT Infrared Deep Sky Survey (UKIDSS)[§]. The first of them contains 2000 selected targets from Sgr A ($\sim 0.5 \times 0.5$ deg) and the second the same number of targets but this time from a particular portion of the Scutum-Centaurus arm. The celestial targets, as well as the resulting clusters, are shown in Figure 5. In scenarios of this kind, bias can also occur but, since the points of interest are scattered over a wider area, it is not as severe as in denser scenarios. Overlapping is also present, but, as expected, superposed regions enclose few points.

5. CONCLUSION AND FUTURE WORK

MIRADAS is a science-driven spectrograph that has been conceived as a GTC common user instrument for many years to come. Consequently, it does not impose severe limitations on the celestial objects fields it can observe. Targets will be selected by scientists mainly considering their sky locations, yielding, in many cases, disperse fields of objects that do not fit in the area of the instrument FOV. In this paper, we have proposed and demonstrated how fields of that kind can be arranged to be observed by MIRADAS. Specifically, it has been proposed a preprocessing step based on cluster analysis that is run before the target allocator. In this step, user-defined objects are grouped so that all those in the same group fit in the FOV area. That is achieved by employing a hierarchical agglomerative clustering algorithm returning also the center of each cluster, which will be used as telescope's pointing information. This technique successfully partitions the data set of interest as well as creates some overlapping areas that might help to redistribute load among different clusters. However, our solution shows bias towards clusters, especially in those located in the remotest areas, with members unevenly distributed. Future work will include modeling the center of each cluster in a way that the load of each group is balanced more homogeneously. We will also explore distinct clustering approaches as well as a solution connecting this clustering step with the target allocator.

ACKNOWLEDGMENTS

This work has been funded by the Spanish MINECO ESP2015-66494-R project, including a percentage from European FEDER funds.

REFERENCES

[1] Eikenberry, S. S., Bennett, J. G., Chinn, B., Donoso, H. V., Eikenberry, S. A., Ettedgui, E., Fletcher, A., Frommeyer, R., Garner, A., Herlevich, M., Lasso, N., Miller, P., Mullin, S., Murphey, C., Raines, S. N., Packham, C., Schofield, S., Stelter, R. D., Varosi, F., Vega, C., Warner, C., Garzón, F., Rosich, J., Gomez, J. M., Sabater, J., Vilar, C., Torra, J., Gallego, J., Cardiel, N., Eliche, C., Pascual, S., Ballester, O., Illa, J. M., Jimenez, J., Cardiel-Sas, L., Galipienzo, J., Carrera, M. A., Hammersley, P., and Cuevas, S., "MIRADAS for the Gran Telescopio Canarias: system overview," in [*Proc. SPIE 8446, Ground-based and Airborne Instrumentation for Astronomy IV, 844657*], McLean, I. S., Ramsay, S. K., and Takami, H., eds., 844657 (Sept. 2012).

[2] Eikenberry, S. S., Murphey, C. H., Mullin, S. A., Bennett, J. G., Raines, S. N., Ackley, K., Stelter, R. D., Garner, A., Sabater, J., Eikenberry, S. A., Chinn, B., Donoso, H. V., Vega, C. S., Gómez, J. M., Torra, J., Herlevich, M. D., Frommeyer, R., and Miller, P., "Demonstration of high-performance cryogenic probe arms for deployable IFUs," in [*Proc. SPIE 9147, Ground-based and Airborne Instrumentation for Astronomy V, 91470X*], Ramsay, S. K., McLean, I. S., and Takami, H., eds., **9140**, 91470X, SPIE, Montreal, Canada (July 2014).

[§]http://www.ukidss.org

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/27/2018
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

[3] Sabater, J., Gómez, J. M., López, M., Torra, J., Raines, S. N., and Eikenberry, S. S., "Kinematic modeling and path planning for MIRADAS arms," in [*Proc. SPIE 9151, Advances in Optical and Mechanical Technologies for Telescopes and Instrumentation, 91515S*], Navarro, R., Cunningham, C. R., and Barto, A. A., eds., 91515S, SPIE, Montreal, Canada (July 2014).

[4] Sabater, J., Riera-Ledesma, J., Torres, S., Garzón, F., Torra, J., and Gómez, J. M., "Target allocation and prioritized motion planning for MIRADAS probe arms," in [*Proc. SPIE 9913, Software and Cyberinfrastructure for Astronomy IV, 99132P (July 26, 2016)*], Chiozzi, G. and Guzman, J. C., eds., 99132P (July 2016).

[5] Aggarwal, C. C. and Reddy, C. K., [*Data Clustering: Algorithms and Applications*], Chapman & Hall/CRC, 1st ed. (2013).

[6] MacQueen, J. and others, "Some methods for classification and analysis of multivariate observations," in [*Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*], **1**, 281–297, Oakland, CA, USA (1967).

[7] De Rham, C., "La classification hiérarchique ascendante selon la méthode des voisins réciproques," *Les Cahiers de l'Analyse des Données* **135**, 144 (1980).

[8] Benzécri, J., "Construction d'une classification ascendante hiérarchique par la recherche en chaîne des voisins réciproques.," *Cahiers de l'analyse des données* **7**(2), 209–218 (1982).

[9] Cormen, T. H., Stein, C., Rivest, R. L., and Leiserson, C. E., [*Introduction to Algorithms*], Computer Science and Intelligent Systems, The MIT Press, 3rd ed. (2009).

[10] Murtagh, F. and Contreras, P., "Algorithms for hierarchical clustering: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2**(1), 86–97 (2012).

[11] Everitt, B. S., Landau, S., Leese, M., and Stahl, D., [*Cluster Analysis*], Wiley Series in Probability and Statistics, John Wiley & Sons, Ltd, Chichester, UK (Jan. 2011).

[12] Leibe, B., Leonardis, A., and Schiele, B., "Robust Object Detection with Interleaved Categorization and Segmentation," *International Journal of Computer Vision* **77**, 259–289 (May 2008).

[13] Lopez-Sastre, R. J., Onoro-Rubio, D., Gil-Jimenez, P., and Maldonado-Bascon, S., "Fast Reciprocal Nearest Neighbors Clustering," *Signal Processing* **92**, 270–275 (2012).

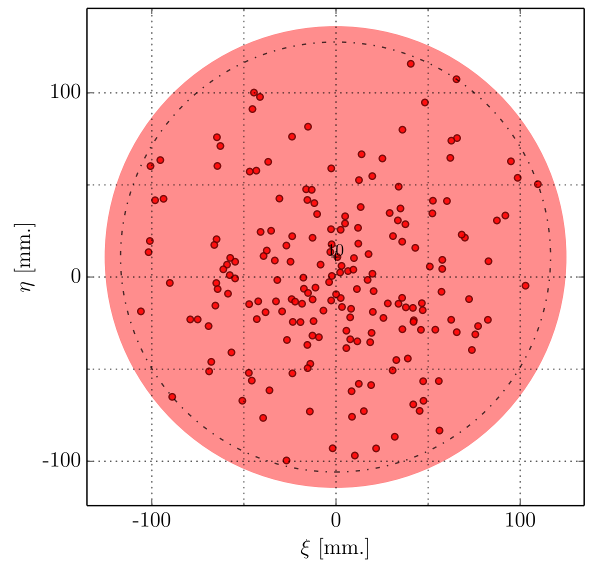
[14] Murtagh, F., "A Survey of Recent Advances in Hierarchical Clustering Algorithms," *The Computer Journal* **26**, 354–359 (Nov. 1983).

[15] Müllner, D., "Modern hierarchical, agglomerative clustering algorithms," *arXiv preprint arXiv:1109.2378* (2011).

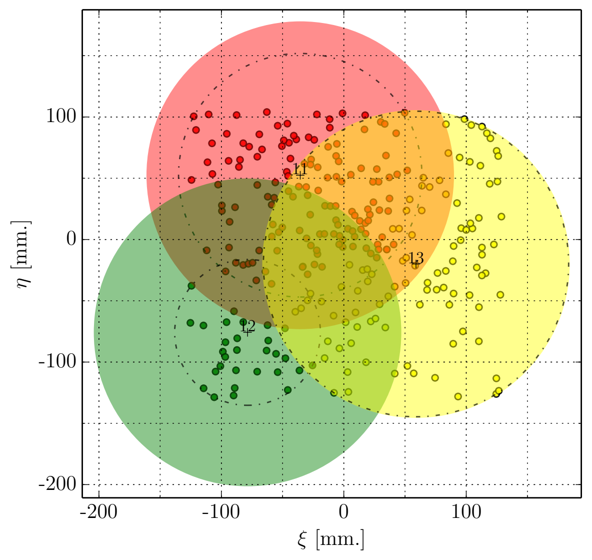
[16] Welzl, E., "Smallest Enclosing Disks (balls and Ellipsoids)," in [*Results and New Trends in Computer Science*], 359–370, Springer-Verlag (1991).

[17] Berg, M. d., Cheong, O., Kreveld, M. v., and Overmars, M., [*Computational Geometry: Algorithms and Applications*], Springer-Verlag, Santa Clara, CA, USA, 3rd ed. ed. (2008).

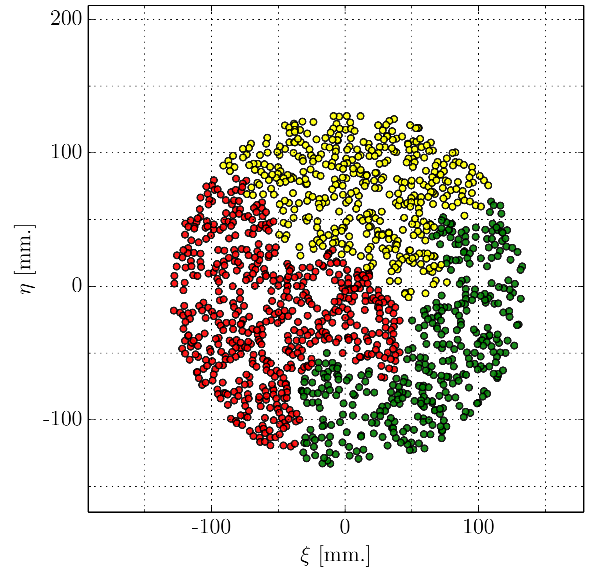
Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/27/2018
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use



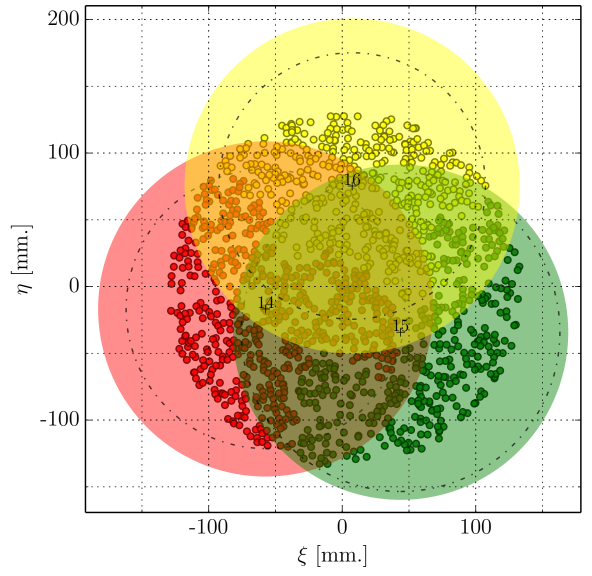
(a) Sky objects (dots) from M13 occupying and area smaller than the MIRADAS FOV (red circle). The algorithm groups all points in a single cluster centered at the black cross.



(b) Targets from Czernik 3 occupying a wider area than the MIRADAS FOV. Here, the agglomerative algorithm returns 3 clusters. The members of the green one are grouped around the cluster center, while those in the yellow cluster are spread over the whole FOV surface. Note that the bounding disk of the yellow cluster has the same diameter as the FOV.



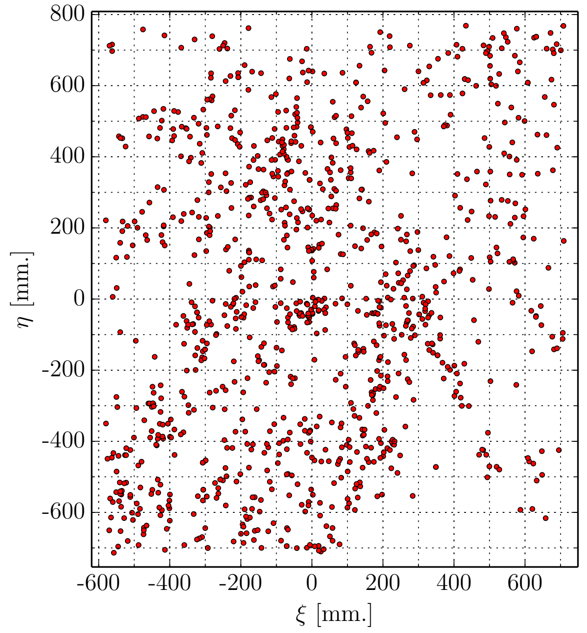
(c) More than 1300 targets from Sgr A occupying a circular area slightly wider than the MIRADAS FOV. For this compact scenario, the clustering algorithm returns three different sets. The dots with the same color belong to the same cluster.



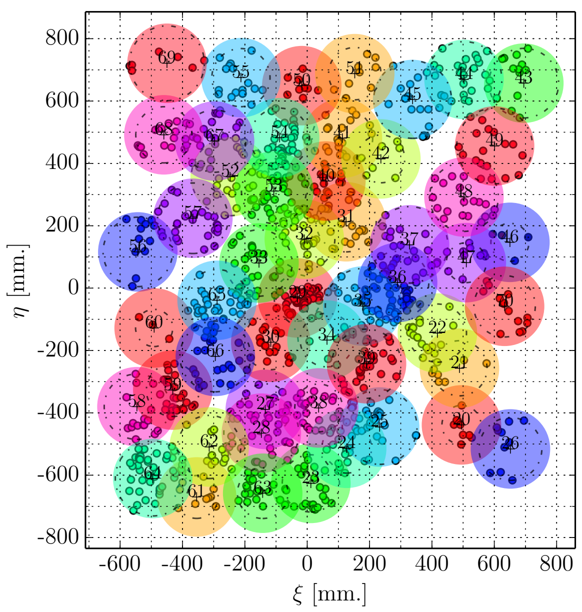
(d) The same points found in Figure 4c, but now the circles specifying the FOV for each cluster are also shown. As can be appreciated, in each cluster there is a bias. The members are not uniformly distributed, leaving a considerable region of the respective FOVs empty.

Figure 4: Clustering results for targets from M13, Czernik 3 and Sgr A.

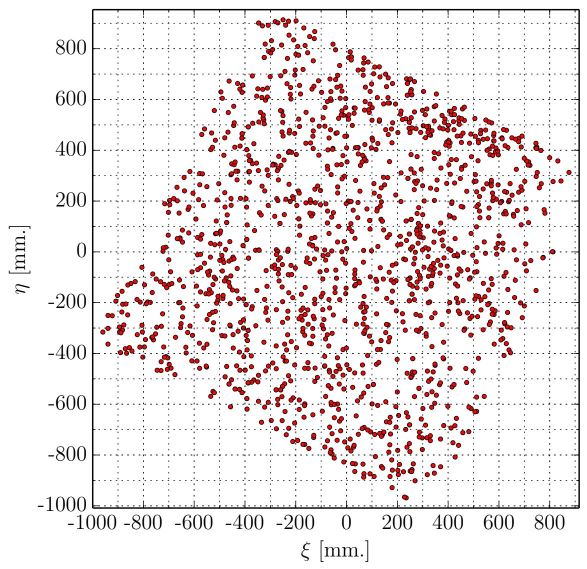
Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/27/2018
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use



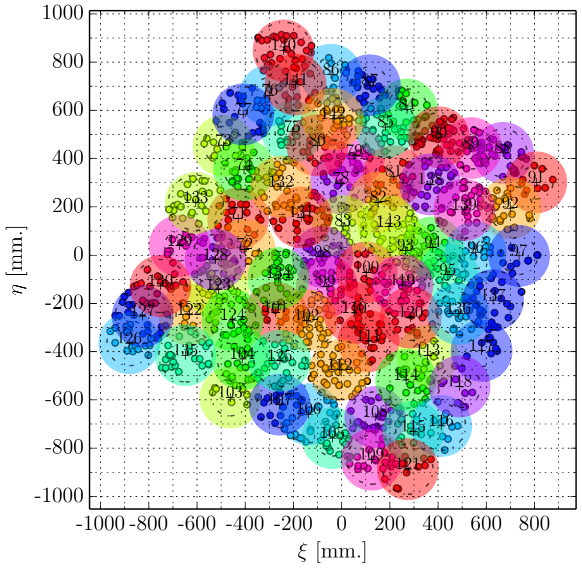
(a) 2000 targets from a wider area of SgrA.



(b) Clusters obtained from the 2000 targets shown in Figure 5a.



(c) 2000 targets from a particular region of the Scutum-Centaurus arm.



(d) Clusters obtained from the 2000 targets shown in Figure 5c.

Figure 5: Clustering results for disperse objects fields belonging to portions of Sgr A and Scutum-Centaurus arm.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/27/2018
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use