

# IACV project F09: Multi-class multi-model fitting via dendrogram cut

Davide Mangano  
Politecnico di Milano

davide1.mangano@mail.polimi.it

Alessandro Messori  
Politecnico di Milano

alessandro1.messori@mail.polimi.it

Giuseppe Paolini  
Politecnico di Milano  
giuseppe.paolini@mail.polimi.it

## Abstract

*Our work aims to improve the T-linkage and Multilink algorithms for fitting multiple instances of a model to noisy data corrupted by outliers. The goal is to replace the dendrogram cut performed by the original methods with another kind provided by an algorithm that performed both dynamic and constant cut, while trying to improve the performance of the two previous models. In particular, we extended both T-linkage and Multilink with a dynamic programming pruning algorithm that allows us to combine the advantages of hierarchical and center-based clustering, and after that we also tried a constant cut to confront the outcomes. Lastly, we performed fine parameter tuning to optimize the model performances and evaluated the results with motion and plane segmentation on the Adelaide dataset. Given the non-deterministic nature of both T-Linkage and Multilink, we evaluated the performance of our algorithms over multiple iterations in order to observe more stable results and reduce the impact of randomness.*

## 1. Introduction

The problem treated in this paper is to analyze the dendrogram produced by agglomerative clustering used in the recovery of multiple geometric structures from noisy data. Once analyzed, the goal is to cut it trying to find the best segmentation and model classes. In particular, our 3 proposed methods defined three different approaches to this problem: from [8] we tried both a dynamic (performs a cut at different heights of the dendrogram) and a constant cut (performs a unique cut on the whole dendrogram), while from [4] we tried an automatic number of cluster selection process based on the minimization of the WCSS via

a probably approximately correct (PAC) framework combined with apriori hypothesis through Bayesian learning.

## 2. Related work

There are some different alternatives available when it comes to multimodel fitting. J-Linkage for example [11], similarly to RANSAC, creates Consensus Sets for each model (set of points with distance from model less than fixed threshold) so to create Preference Sets to define points (set of models they have given consensus to). In order to create clusters, the mechanism takes into account models matched by each point to understand points belonging to same clusters. While this is done in a binary approach in J-Linkage, T-Linkage [6] introduces a variation constituted by a continuous relaxation of the binary Preference Set called the Preference Function of a point. This allows the algorithm to be a generalized version of J-Linkage and to increase accuracy on the residuals, in a way that better expresses preferences of a point. Another keypoint of the differences between J-Linkage and T-Linkage are the calculations on the similarity between two clusters: the first one uses Jaccard distance that, given as input two preference sets, outputs a degree of similarity between 0 and 1; the second one exploits Tanimoto distance which instead operates with the preference function with 1 as cutoff value (in which case it means that preference values are orthogonal).

More recent approaches such as Multilink [7] leverage single-linkage clustering inside a multi-class preference space.

In Multilink the decision of joining two clusters or not is taken by performing on-the-fly model fitting on both the separated and merged groups, and then merging the clusters only if the GRIC [10] score increases when compared to

the separated models. The dendrogram is then cut automatically when there aren't any possible unions among clusters that would increase the GRIC score, computed with the following formula, taken from [7].

$$g_k(U) = \sum_{x_i \in U} \rho \left( \frac{\text{err}(x_i, \vartheta_k(U))}{\sigma} \right)^2 + \lambda_1 d|U| + \lambda_2 \mu \quad (1)$$

### 3. Proposed approach

The main objective of our work is to explore and compare automatic methods for the dendrogram cut task with respect to the approaches utilized by existing preference selection algorithms, which are usually pretty basic. With this objective in mind, our main task was to figure out how we could integrate the automatic cut techniques we found in our literature review with the existing methods of T-Linkage and Multilink.

#### 3.1. Gmart: Dynamic dendrogram Cut

Thanks to dynamic dendrogram pruning, we are able to combine advantages of both hierarchical and center-based clustering. In G. Marti algorithm [8], our aim is to perform the best cut possible:

$$f(T, k) = \min_{0 < k_l < k} f(T_l, k_l) + f(T_r, k - k_l) \quad (2)$$

$$s.t. : k_l + k_r = k \quad (3)$$

where T is the dendrogram and T<sub>l</sub> and T<sub>r</sub> respectively left and right subtrees of the dendrogram, while k is the number of cluster that is our goal to extract from T to obtain a flat clustering. With dynamic programming, this can be done in a time that is polynomial:

$$O(k \cdot n \log n) \quad (4)$$

To recap, we will have k cluster in the current node T: then if k = 1 all of its children are in the same cluster, otherwise we have to allocate them in subtrees such that data points concerned by T<sub>l</sub> go in k<sub>l</sub> clusters and viceversa. This is computed by function *compute\_dynccut* in a bottom-up fashion.

#### 3.2. Gmart: Constant dendrogram Cut

As an alternative to the dynamic cut algorithm, we also integrated to T-linkage and Multilink the constant dendrogram cut version.

The latter works by finding a minimum threshold r such that the cophenetic distance (the height of the dendrogram where the two branches that include the two objects merge into a single branch) between all pairs of points in the same cluster is less than or equal to r and that the number of clusters created are not greater than NUMBER\_OF\_CLUSTER.

### 3.3. PAC Bayesian dendrogram Cut

The PAC Bayesian Cut proposed in [5] builds upon the dynamic cut method presented in the previous section, but adds an automatic estimation of the number of clusters with Bayesian and probably approximately correct learning [2]. The basic idea is to find the model with the optimal cut by selecting the parameter configuration which yields the lowest bound of the likelihood in the Bayesian equation:

$$\log p(D | M) \geq \int \log p(D | M, \theta) q(\theta) d\theta - D_{KL}(q(\theta) || p(\theta | M)) \quad (5)$$

Where (D, M, θ) represent respectively the observed data, the Model and its parameters.

This equation in our specific case can be simplified to:

$$R_{in}(D | Z^k) - \frac{1}{\lambda} \log p(Z^k | M) \quad (6)$$

Where Rin is the total WCSS given by the model configuration with k clusters.

We also need to define an apriori distribution for the Bayesian learning, we decided to go with the Chinese restaurant process, which is one of the most common formulas used for depicting the distribution of cluster labels:

$$p(Z_i^k = z | Z_{1:i-1}^k, M, \alpha) = \begin{cases} \frac{\alpha}{i-1+\alpha}, & z \notin Z_{1:i-1}^k \\ \frac{n_z}{i-1+\alpha}, & \text{otherwise} \end{cases} \quad (7)$$

#### 3.4. T-Linkage Complete dendrogram

In this subsection we describe the experiment we performed by combining T-Linkage with the method provided by G. Marti [8] to perform a dynamic cut of the dendrogram. In our work, the goal as has been already mentioned was to improve the cut of the dendrogram to reduce the Mean Error. In this case, starting from the work available online [1] we observed the functioning to see where to introduce our alternative dendrogram's cut method; we had to modify part of the original code in order to allow T-Linkage to generate a complete dendrogram without stopping the creation of it when the tanimoto distance between two clusters that are trying to combine gets larger or equal to one. In this way we obtained a complete dendrogram in the form of a suited matrix to give as input to G. Marti and PAC Bayesian cutting algorithms.

#### 3.5. Multilink Complete dendrogram

Just like with T-Linkage, Multilink also can not be natively integrated with dynamic cut methods since it does not produce a complete dendrogram. However, while this issue can be easily solved in T-Linkage by removing the constant cut at distance 1, the situation is more complicated with Multilink, since in the original implementation after

each rejected merge the distance among the two selected clusters is set to infinity, and thus it is not possible to obtain a complete dendrogram once the Multilink clustering has completed.

Our solution to this problem aims to complete the dendrogram in a bottom-up fashion: since we cannot use the on-the-fly fitting + GRIC comparison technique anymore, we start from the clusters generated by Multilink and then perform a simple hierarchical clustering based on tanimoto distance until we reach the root of the dendrogram. This method allows to obtain a complete tree structure compatible with the automatic cut techniques while at the same time leaving the original Multilink aggregations untouched.

To implement this modified version of Multilink it's necessary to add a second distance matrix, that is used to store the tanimoto distances among all clusters which will not be set to infinity after each merge rejection.

Thanks to this auxiliary matrix, we are able to perform the hierarchical clustering on the partial dendrogram once the original Multilink iteration has completed.

### 3.6. Automatic Cut Integration

Now that we have modified both T-Linkage and Multilink in order to obtain a complete dendrogram, the integration with the Gmart and PAC Bayesian algorithms is pretty straightforward: since they just need any complete, well formed dendrogram as input to work, we just need to feed the output tree of T-Linkage/Multilink to our cutting techniques to obtain as a result our estimated clusters.

The small clusters are then pruned in order to increase accuracy, how we estimated the best threshold value for the cluster pruning is discussed in the following section.

### 3.7. Illustrations about our results with G. Marti cut

In Figure 1, Figure 2, Figure 3 and Figure 4 we present some results we obtained while performing our cut. As it can be seen in the captions, our model fitting performs better than original T-Linkage (for Figure 1 and 2) and Multilink (for Figure 3 and 4).

## 4. Experiments

**Datasets.** We used as dataset AdelaideRMF [6], which is a dataset for geometric model fitting used for both homography and fundamental matrix estimation, available online [3]. It is composed of a set of image pairs on which SIFT matching allowed to manually label keypoint correspondences. In order to test our algorithms we used 38 images from the dataset, half of them was used for motion segmentation and the other half for plane segmentation.

**Experiments setup.** After observing initial results, we decided to do some fine parameter tuning with 2 values in

---

### Algorithm 1 Complete dendrogram Multilink

---

**Input :** X data,  $\{\Theta_k\}_{k=1,\dots,K}$  model classes,  $\epsilon$  inlier threshold,  $\lambda_1, \lambda_2, GRICparameters$

**Output :** *dendrogram tree*  $Dt$  representing the complete clustering process

```

1: Sample hypotheses  $H = H_1 \cup \dots \cup H_K$ ;
2: Compute preferences  $p(x_i, H) \forall x_i \in X$ ;
   // Clustering Starts
3: Put each point  $x_i$  in its own cluster  $\{x_i\}$ ;
4: Compute inter-cluster distances  $d$  from matrix  $D_1$ ;
5:  $D_2 = \text{copy}(D_1)$ ;
6: while  $\min d \in D_1 < \infty$  do
7:   Find clusters  $(U, V)$  with the min distance;
8:   for  $k = 1$  to  $K$  do
9:     Fit a model  $\vartheta_k(U)$  to  $U$  and compute  $g_k(U)$ ;
10:    Fit a model  $\vartheta_k(V)$  to  $V$  and compute  $g_k(V)$ ;
11:    Fit a model  $\theta_k(U \cup V)$  to  $(U \cup V)$  and compute
        $g_k(U \cup V)$ ;
       // Test Merge Condition
12:   if  $\exists k : g_k(U \cup V) \leq g_k(U) + g_k(V) \forall k$  then
13:     merge  $U$  and  $V$ , the structure is  $\theta_k(U \cup V)$ ;
14:     update inter-cluster distances in matrix  $D_1$ ;
15:     update inter-cluster distances in matrix  $D_2$ ;
16:     update dendrogram tree  $Td$ ;
17:   else //Set distance to infinity in matrix  $D_1$ 
18:      $D_1(U, V) = +\infty$ ;
19: // — End of the original Multilink algorithm —
20:
21: /* To complete the dendrogram we apply hierarchical
   clustering on the auxiliary matrix  $D_2$  */
22: while  $\text{size}(D_2) > 1$  do
23:   Find clusters  $(U, V)$  with the min distance;
24:   merge  $U$  and  $V$ , the structure is  $\theta_k(U \cup V)$ ;
25:   update inter-cluster distances in matrix  $D_2$ ;
26:   update dendrogram tree  $Td$ ;

```

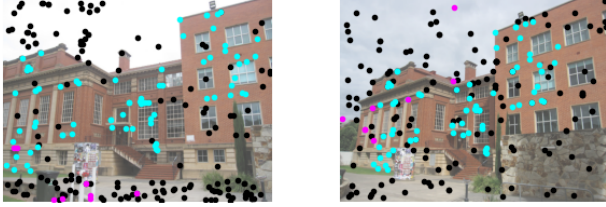
---

Gmart: NUMBER\_OF\_CLUSTER (the total number of clusters we create at first) and OUTLIER\_TRESHOLD (the minimum number of points in a cluster to be defined as such, otherwise it is moved in a general cluster containing all the outliers).

After this, we observed some randomness in the ME in output, so in order to clarify better our results we decided to execute multiple iterations of the algorithm for each image to observe its average behaviour.

**Parameters Tuning** We performed parameters tuning on two parameters, specifically we assigned values in a range from 60 up to 8 for OUTLIER\_TRESHOLD and from 100 up to 2 for NUMBER\_OF\_CLUSTER. During the experiments on the dataset we noticed that depending on the pa-

Figure 1. T-Linkage Estimation has ME = 0.153%  
, T-Linkage + Gsmart Flat Estimation has ME = 0.120%  
barrsmith - T-Linkage Estimation



barrsmith - TLinkage + Gsmart Flat Estimation

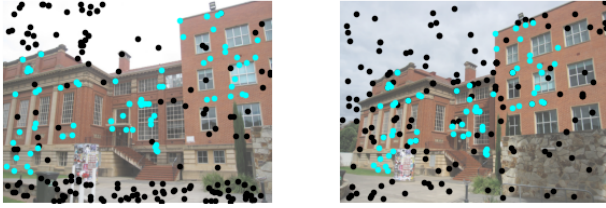
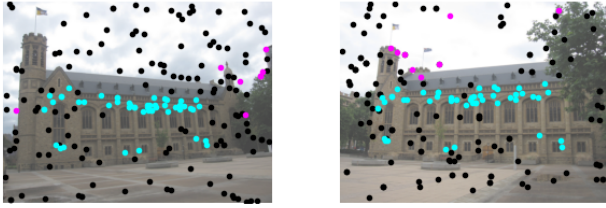


Figure 2. T-Linkage Estimation has ME = 0.050%  
T-Linkage + Gsmart Flat Estimation has ME = 0.025%  
bonython - T-Linkage Estimation



bonython - TLinkage + Gsmart Flat Estimation

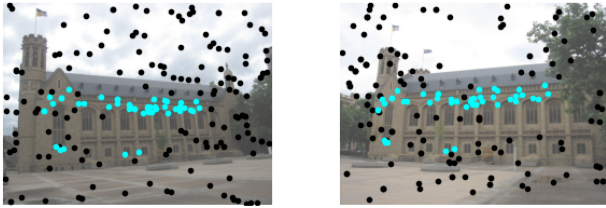
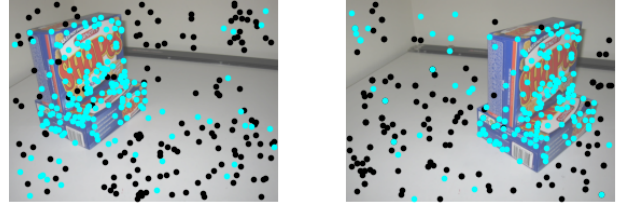


Figure 3. Multilink Estimation has ME = 0.106%  
Multilink + Gsmart Dyn Estimation has ME = 0.003%  
biscuit - Multilink Estimation



biscuit - Multilink + Gsmart Dyn Estimation

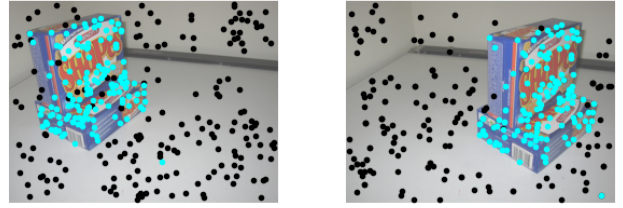
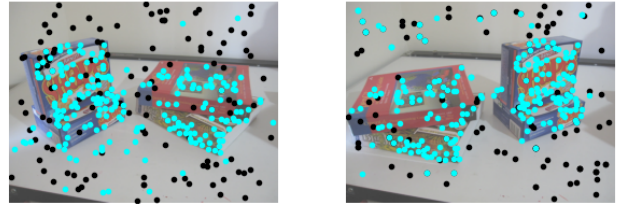
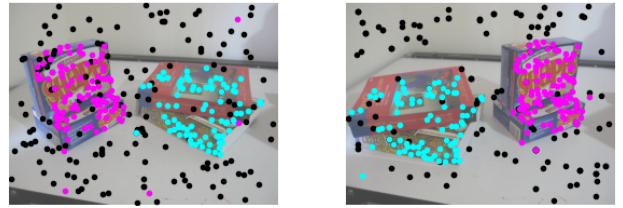


Figure 4. Multilink Estimation has ME = 0.34%  
Multilink + Gsmart Dyn Estimation has ME = 0.023%  
biscuitbook - Multilink Estimation



biscuitbook - Multilink + Gsmart Dyn Estimation



rameters used we could reduce the mean error in plane segmentation or motion segmentation. We explored the parameter spaces with intervals of 10 and 5 units for both hyperparameters and arrived at a pair of optimal values for the provided dataset as shown in the figure.

**Evaluation** As T-linkage and Multilink have a small dose of randomness in the final results due to their implementation, to best evaluate our algorithms we tried to decrease the imprecision due to the random nature of the code by performing for each sample of the dataset 5 iterations and

used the mean error of the ME and the standard deviation produced in each of these for each sample as metrics.

**Results and discussion.** Our best results after performing parameter tuning are summarized in Tables 1 and 2. As can be seen, the integration of T-linkage with the algorithm for constant dendrogram cutoff performs better than the original T-linkage in plane segmentation, while both integrations of Multilink with G. Marti algorithm (with both dynamic and constant cutoff) produce a better result than the original version in plane and motion segmentation.



Unfortunately, we were not able to improve the performance of the original model with the PAC Bayesian cut method.

Method	Mean Error	Std
T-link	0.071	0.053
T-link + Gmart Dyn	0.1977	0.1355
T-link + Gmart Cos	0.1370	0.1131
T-link + PAC Bayesian	0.5296	0.0446
M-link	0.346	0.1311
M-link + Gmart Dyn	<b>0.1442</b>	0.1452
M-link + Gmart Cos	0.1854	0.1779

Table 1. Comparison Table on Motion Segmentation

Method	Mean Error	Std
T-link	0.2605	0.1456
T-link + Gmart Dyn	0.4920	0.1572
T-link + Gmart Cos	<b>0.2044</b>	0.1831
T-link + PAC Bayesian	0.5218	0.0679
M-link	0.4191	0.1175
M-link + Gmart Dyn	0.3417	0.117
M-link + Gmart Cos	0.2396	0.0994

Table 2. Comparison Table on Plane Segmentation

## 5. Conclusion

In this project we experimented with 3 different kinds of automatic dendrogram cuts techniques by integrating them to the preference based multi-model fitting algorithms T-Linkage and Multilink, testing them on motion and plane segmentation on the Adelaide Dataset.

The automatic cut approaches proposed by G. Marti managed to improve both T-Linkage and Multilink in many images of both subsets of the Adelaide Dataset, while the PAC Bayesian approach didn't manage to produce significant improvements over the original techniques.

There are still more cutting techniques to be explored[9], and with more extensive fine tuning techniques it could be possible to find even better hyper parameters than the ones we found.

In conclusion, we have proved the potential of automatic dendrogram cutting techniques in a preference selection scenario and built the basis to future further improvements in the field.

## References

- [1] Antonio Maria Rizzo. Python implementation of T-Linkage, 2014. <https://github.com/rizzoantoninomaria/tlinkage>. 2
- [2] David Haussler. Probaby Approximated Correct Learning, 1990. <https://bit.ly/3TSfZ3d>. 2
- [3] Gerard Sanroma. Adelaide dataset, 2015. <https://osf.io/gb5yc/>. 3
- [4] Jerry Lin. Automatic Dendrogram Cut, 2021. <https://towardsdatascience.com/automatic-dendrogram-cut-e019202e59a7>. 1
- [5] Jerry Lin. Automatic Dendrogram Cut - PAC Bayesian in Practice, 2021. <https://towardsdatascience.com/automatic-dendrogram-cut-e019202e59a7>. 2
- [6] Luca Magri, Andrea Fusiello. T-Linkage: a Continuous Relaxation of J-Linkage for Multi-Model Fitting, 2014. <https://bit.ly/3AI0XEw>. 1, 3
- [7] Luca Magri, Filippo Leveni, Giacomo Boracchi. MultiLink: Multi-class Structure Recovery via Agglomerative Clustering and Model Selection, 2021. <https://bit.ly/3wTpBB7>. 1, 2
- [8] G. Marti. How and where should you cut a dendrogram?, 2017. <https://gmarti.gitlab.io/ml/2017/05/12/cut-a-dendrogram.html>. 1, 2
- [9] Peter Langfelder, Bin Zhang. Hybrid Dynamic Tree Cut method, 2007. <https://github.com/kyleessmith/dynamicTreeCut>. 5
- [10] Philip H. S. Torr. Model Selection for Two View Geometry:A Review, 1999. pages 277–301. Springer Berlin Heidelberg, Berlin, Heidelberg. 1
- [11] Roberto Toldo, Andrea Fusiello. Robust Multiple Structures Estimation with J-Linkage, 2008. [https://link.springer.com/chapter/10.1007/978-3-540-88682-2\\_41](https://link.springer.com/chapter/10.1007/978-3-540-88682-2_41). 1