Tomographic reconstruction of objects comprising of heterogeneous structures

By Arunabh Ghosh

Advisor: Professor Ajit Rajwade

Motivation

- The primary motivation for tackling this problem comes from the field of cryo-electron microscopy which has become an important technique for determining the geometry of a biological molecule.
- It is often the case that virus particles can be grouped into a number of discrete classes depending on their geometric forms, sizes, pH values, etc.
- If the case of a single object reconstruction is well studied, the heterogeneous case is an active research field. In this thesis project, I have developed algorithms to automatically detect the number of conformations from the set of projections and accurately reconstruct all of them.
- Several strategies have been proposed for classification of heterogeneous projection data. The most potent one may be supervised classification which requires prior structural knowledge about the sample heterogeneity. Here the dependency on a priori available information seriously limits the general applicability of this approach.
- Secondly, some strategies use a maximum-likelihood based approach where the number of conformations is specified by the user. This specification by the user, sometimes leads to important conformations being missed by biologists.

Preprocessing the projections

- In Cryo-EM, most biological specimens are extremely radiosensitive, so they must be imaged with low-dose electron beams which leads to extremely high amounts of noise in the projections.
- For the single reconstruction case, we simply cluster the projections as the averaging greatly reduces the noise. In this case however, we need to cluster the projections such that not only are the orientations similar but they belong to the same class as well.
- If a cluster, comprises of projections belonging to two or more classes, the cluster essentially cannot be used to construct either one of the classes. This makes the purity of the cluster highly important.
- To achieve this task we use a variant of agglomerative clustering algorithm called single-linkage clustering.
- At each step, the two clusters separated by the shortest distance are combined. In single-linkage clustering, the distance between two clusters is determined by a single element pair, namely those two elements that are closest to each other.
- The two elements that are most likely to be close are the projections taken at similar angles. If these two projections come from different classes, the heterogeneity would make a difference and the two clusters would not be merged. On the other hand, if the projections come from the same class, they would almost be the same and the clusters would be merged.

Preprocessing the projections

- We compare our baseline model K-means clustering , against two variants of Hierarchical Agglomerative Clustering, one in which the *single-linkage* criteria is used and in the other, *average-linkage* criteria is used.
- Thus we can see that the *single-linkage* criteria clusters the projections such that 99.78% of the projections belong to a single class of the object.
- Now that we have created pure clusters, we denoise those clusters using a PCA based denoising algorithm. These clusters are now ready to be classified into their respective classes.

Clustering Algorithm	**Average Cluster Purity**
K-means clustering	81.09%
Average Linkage	82.44%
Single Linkage	99.78%

Classifying the clustered projections

- Now that we have obtained pure clusters, our task is to classify the clusters into their respective classes.
- What we observe from the difference classes of the object is that due to heterogeneity the average value of each class of object is different.
- Using the Helgasson Ludwig Consistency Conditions (HLCC) we can translate this into an equivalent condition in the projection space.
- Given below is the HLCC condition relating the image moments to the projection moments.

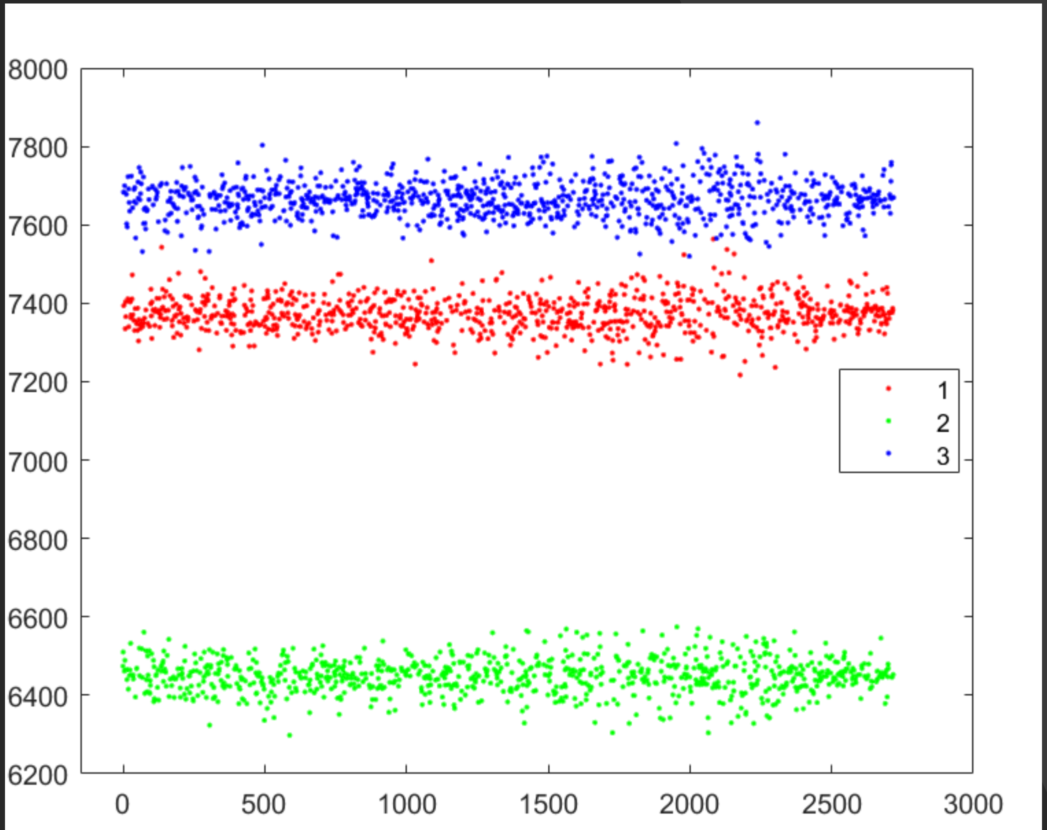
$$m_{\theta_i}^{n} = \sum_{j=0}^{n} \binom{n}{j} (cos\theta_i)^{n-j} (sin\theta_i)^{j} v_{n-j,j}$$

Classifying the clustered projections

- For n=0, the above condition becomes:

$$m_{\theta_i}^{0} = v_{0,0}$$

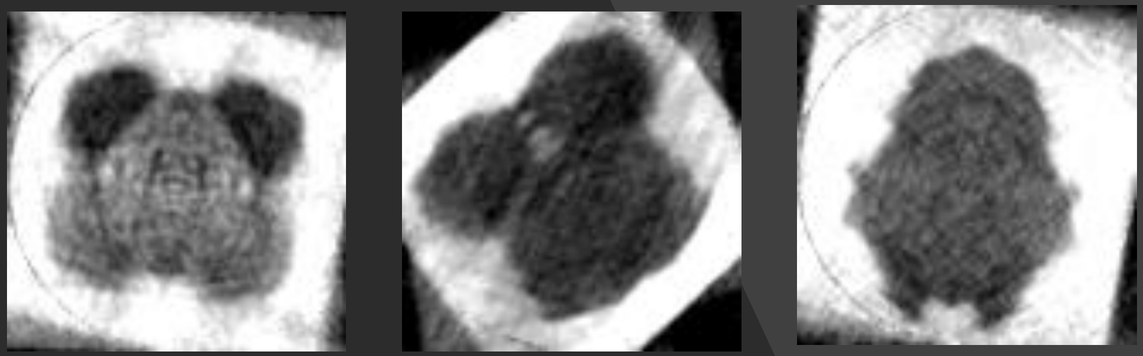
- What this tells us that the zeroth moment of a projection (LHS) is equal to the average of the object it belongs to (RHS).
- This means that, all projections belonging to a single class will have the same zeroth-order moment. Therefore we can use the zeroth order moment as a statistic to classify the clusters into their respective classes. This is shown on the graph beside.
- Thus, we segregate the clusters by their zeroth-order moment and proceed to independently construct each of the object.



Independent reconstruction

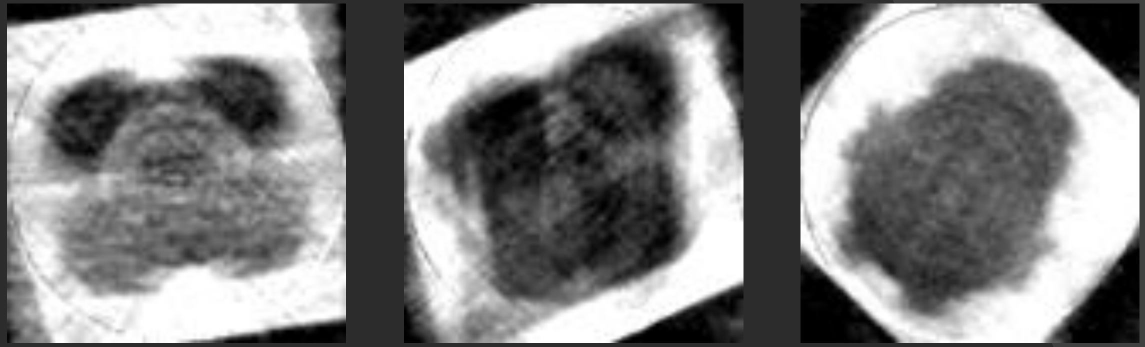
- After classifying the projection clusters, we have essentially reduced the problem to the case of single object reconstruction. We have already have developed an algorithm for this case. The summary is stated below for the sake of completeness.
- Our algorithm clusters similar projections together, uses a moments based approach to obtain an initial estimate for the orientations and finally optimizes for the structure of the unknown object along with a refinement of the viewing parameters using an alternate minimization scheme.
- For a detailed overview of the algorithm, refer to the paper here.

20000 Projections, 15% noise



23.34% 28.03% 18.97%

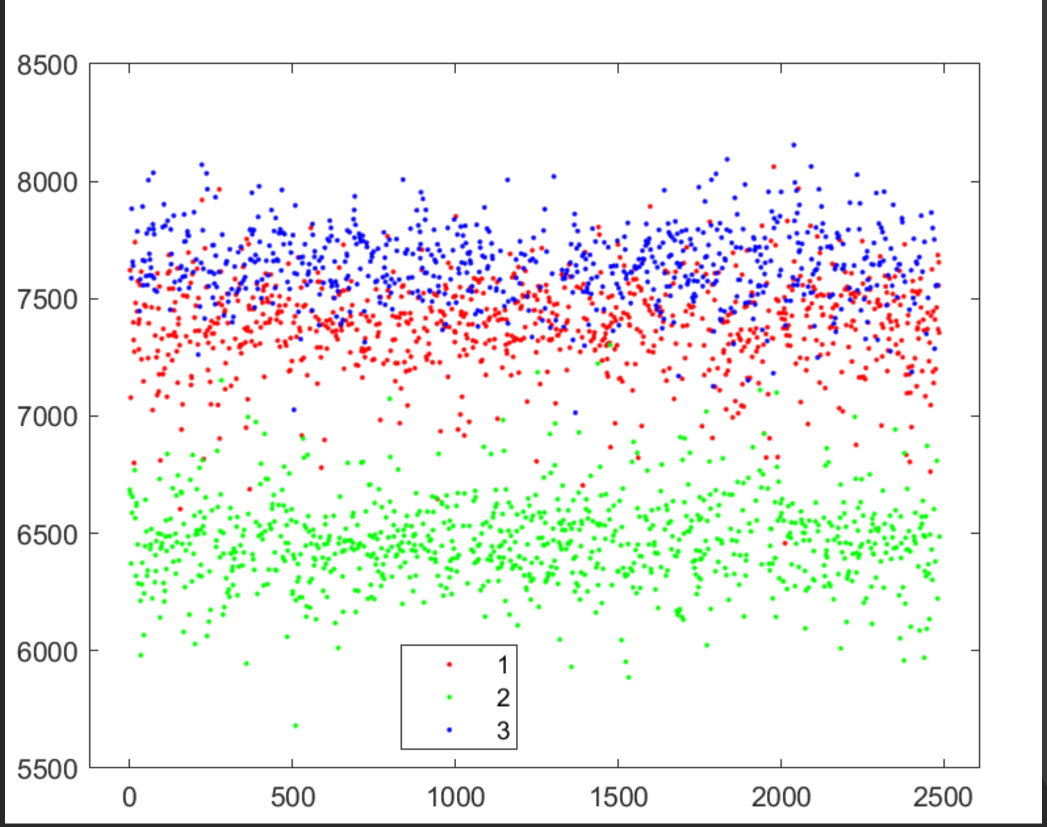
20000 Projections, 25% noise



25.53% 33.38% 19.41%

Disadvantages of the zeroth-moment clustering approach

- In cases where the heterogenous objects are similar, or in the presence of pathological amounts of noise, clustering just on the basis of the zeroth-moments leads to the misclassification of certain clusters.
- Over here, about 10% of the projection clusters are misclassified. This is actually a significant percentage as classifying one projection cluster incorrectly implies, all the projections associated with that cluster are misclassified.
- This results in the reconstruction being significantly corrupted.

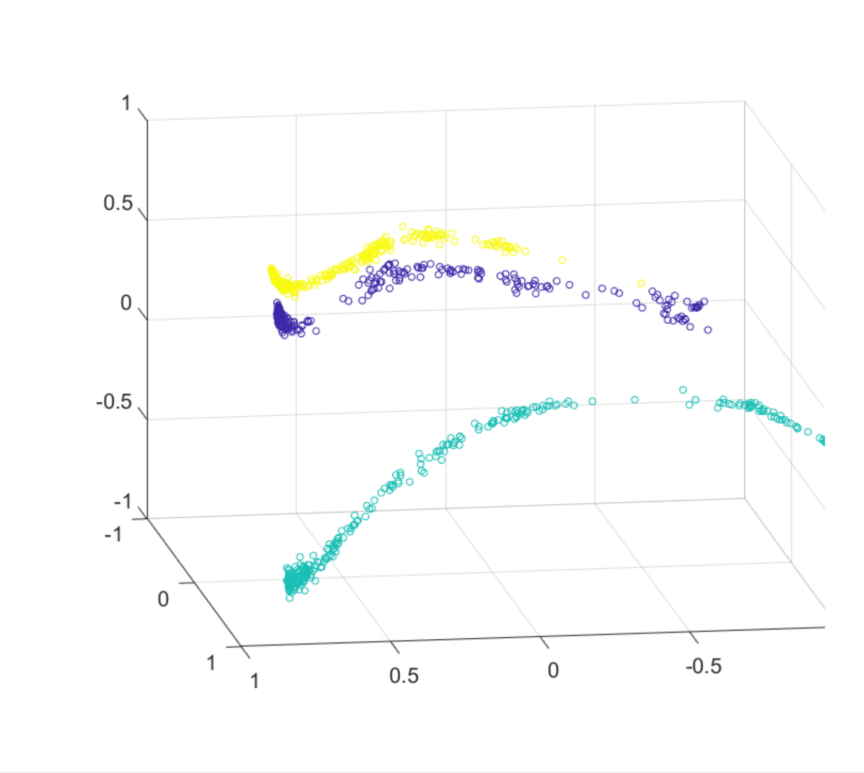


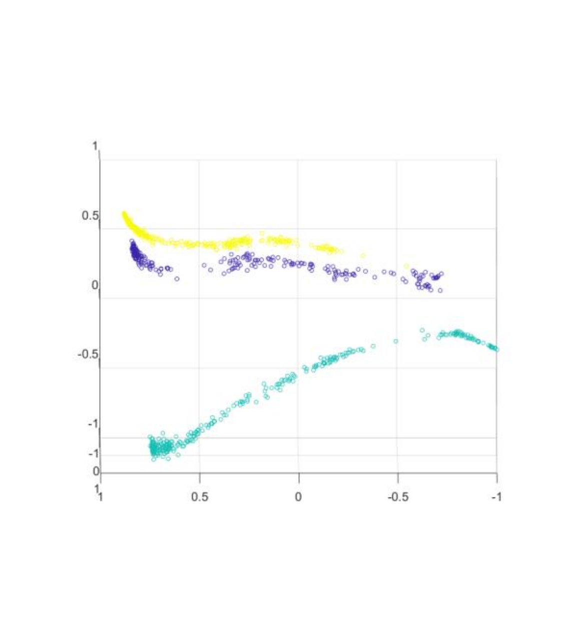
Robust Classification

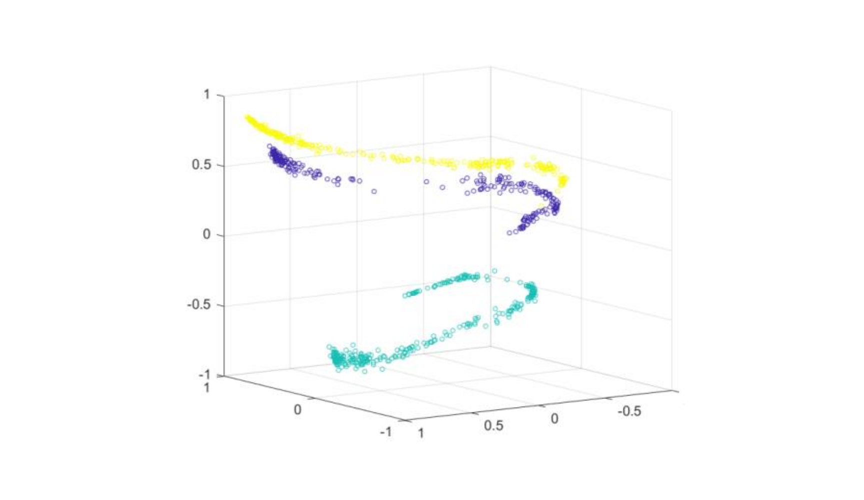
- What we need is a robust classification scheme, that takes into account not just one parameter, but characteristics of the entire set of projections.
- For this we introduce a graph Laplacian-based algorithm. A Laplace-type operator is constructed on the data set of projections, and the eigenvectors of this operator reveal the classes present in the projections.
- Graph Laplacians are widely used in machine learning for dimensionality reduction and spectral clustering. Briefly speaking, we construct an N×N weight matrix related to the pairwise projection distances, followed by a computation of its first few eigenvectors. The top eigenvectors when plotted reveal the classes of the projections in a manner to be later explained. More importantly, the graph Laplacian incorporates all local pieces of information into a coherent global picture, eliminating the dependence of the outcome on any single local datum. Small local perturbations of the data points have almost no effect on the outcome.

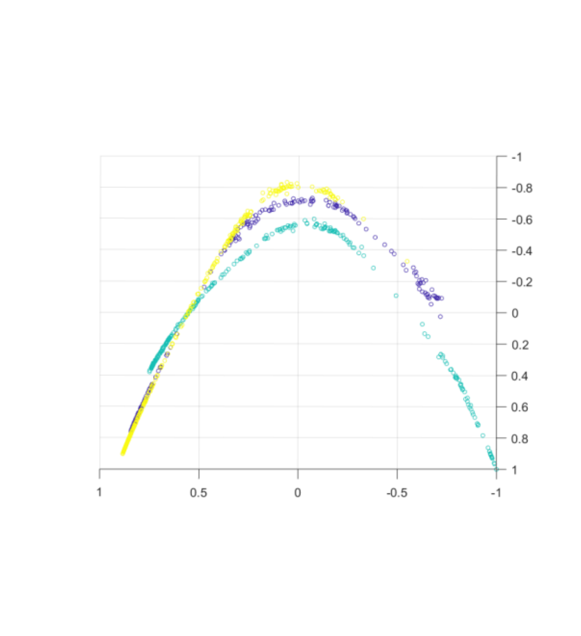
Graph-Laplacian based clustering

- We follow the same pre-processing step as mentioned before to obtain the pure clusters. We then reduce the dimensionality using Graph-Laplacians and plot just the top three values.
- Shown on the side are the results of the dimensionality reduction in the case of three classes.
- Indeed we are able to see three well defined clusters in the shape of strips!



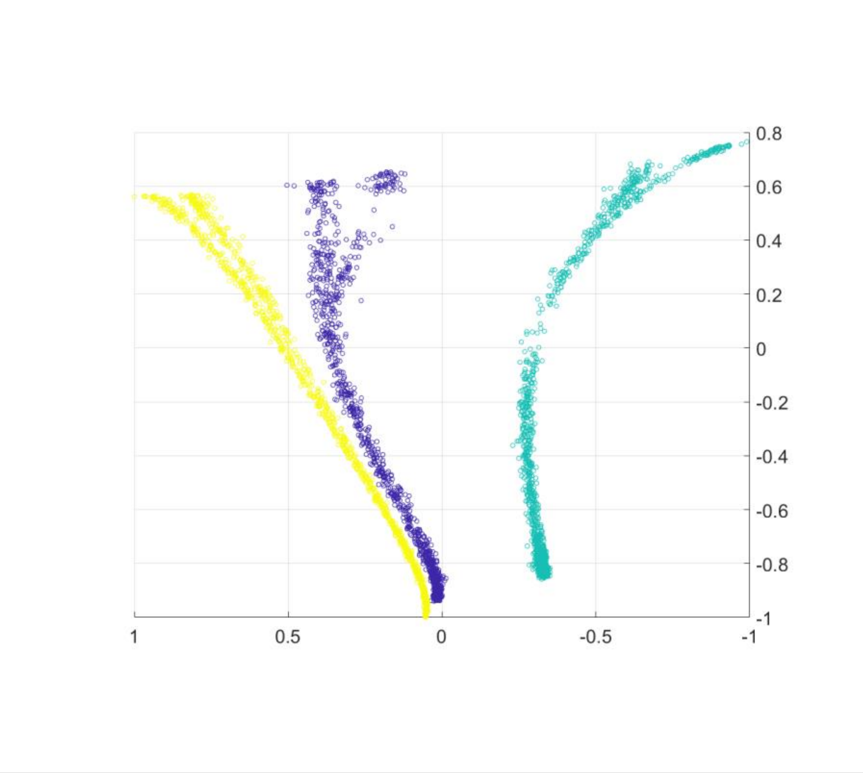


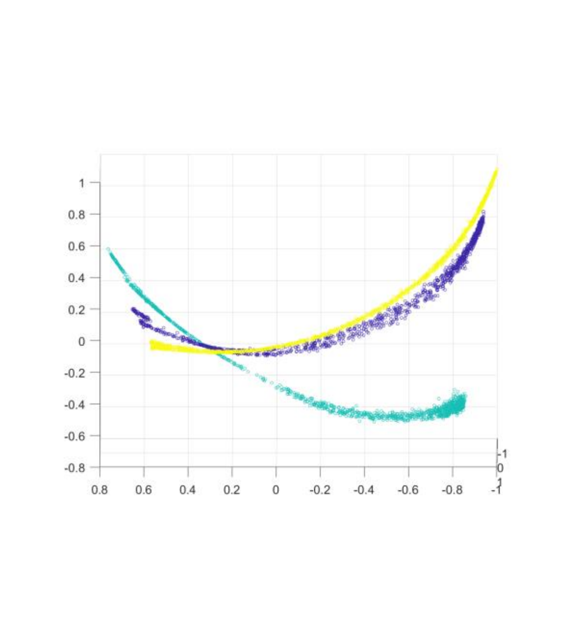


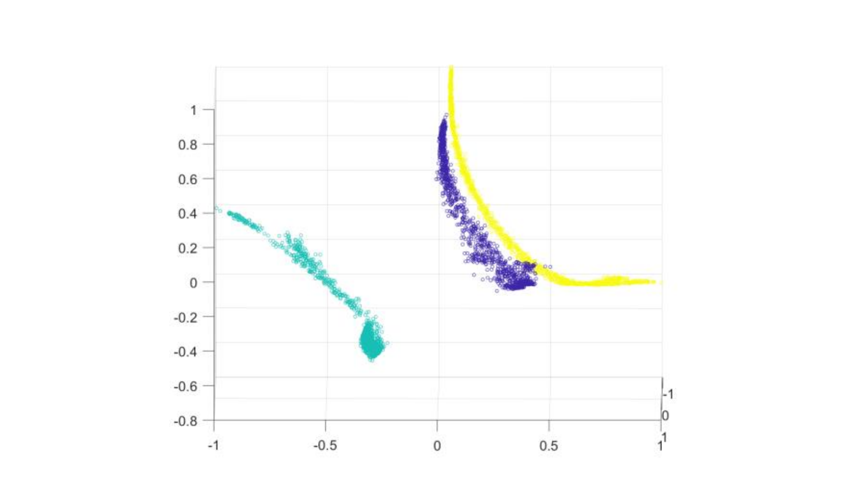


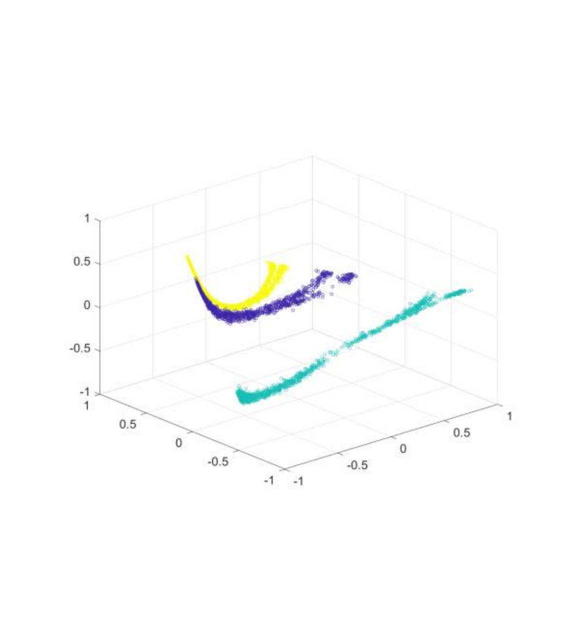
Graph-Laplacian based clustering

- Even in the case of noisy projections, where we could see the zeroth-order moment based clustering fail, the graph Laplacian based dimensionality reduction clearly distinguishes the clusters as shown on the side.









Identifying the clusters

- Now that we have able to visualize the clusters, our the only step remaining is to identify the clusters automatically using a clustering algorithm.
- This turns out to be a surprisingly difficult task, as the clusters are of unusual shapes in the form of long thin strips.
- A number of approaches have been a tried, but a robust automated clustering algorithm is yet to be discovered.
- Given next are some of the approaches we tried, and also the pros and cons of each one of them.

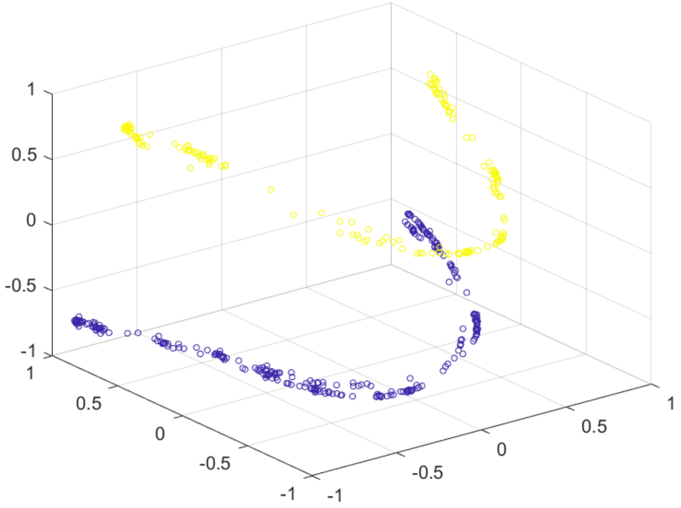
Kernel K-means

- This algorithm applies the same trick as k-means but with one difference that here in the calculation of distance, kernel method is used instead of the Euclidean distance.e same trick as k-means but with one difference that here in the calculation of distance, kernel method is used instead of the Euclidean distance.
- The transformation of non-linear data into a higher dimensional feature space increases the probability of the linear separability of transformed data.

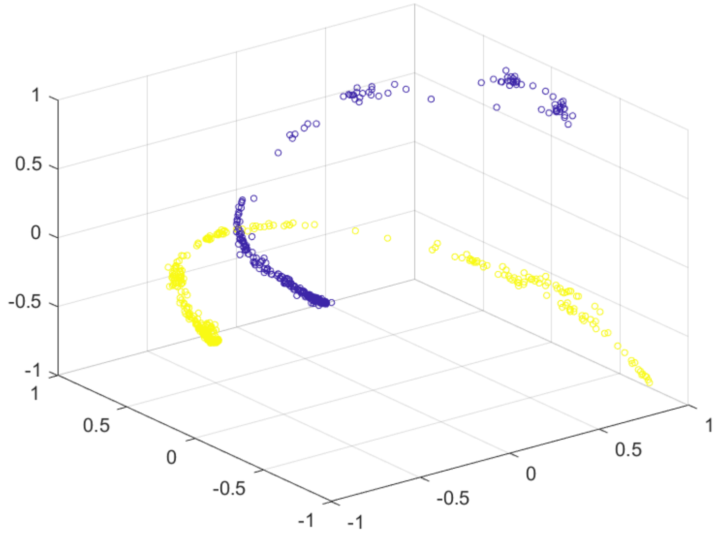
Disadvantages:

- In the case of high noise, Kernel K-means is not able to effectively identify the clusters, even when they are plainly visible the human eye.

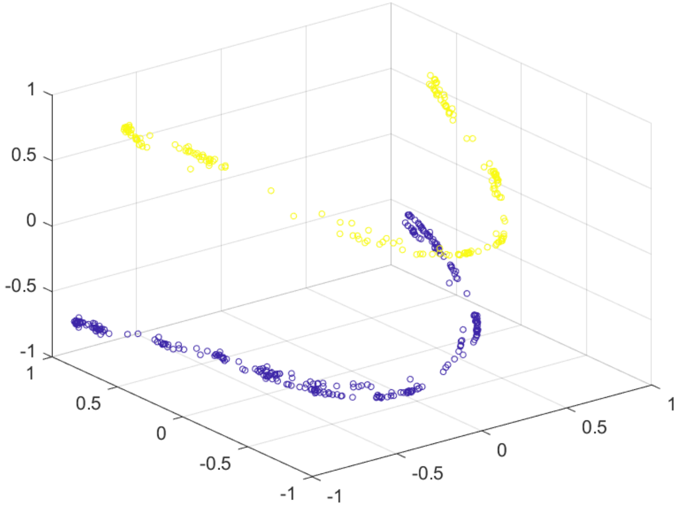
Original Clusters 10% noise



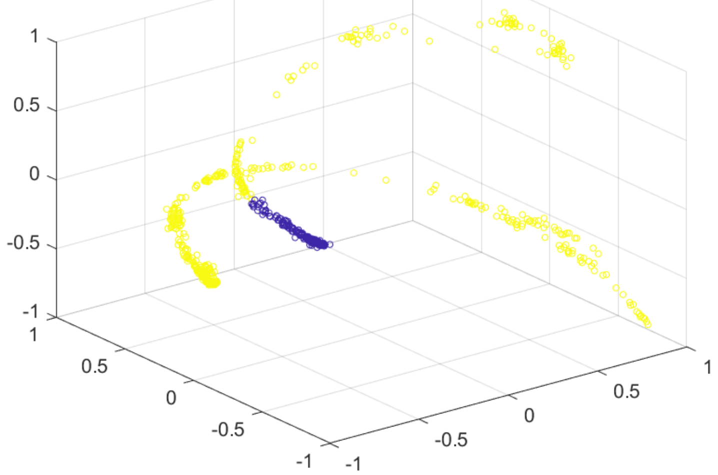
Original Clusters 20% noise



Successful Classification 10% noise



Unsuccessful Classification 20% noise



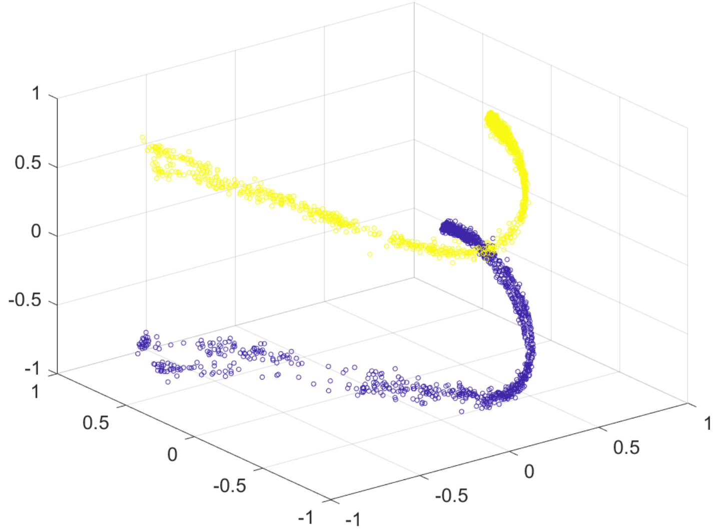
Multi-RANSAC based algorithm

- Random sample consensus (RANSAC) is an iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers, when outliers are to be accorded no influence on the values of the estimates.
- Here we observe that the classes are in shape of smooth curves. Therefore we try to fit multiple polynomials to identify the clusters shown over here.
- To obtain reliable clustering, we use the zeroth-moment based estimate as an initialization.

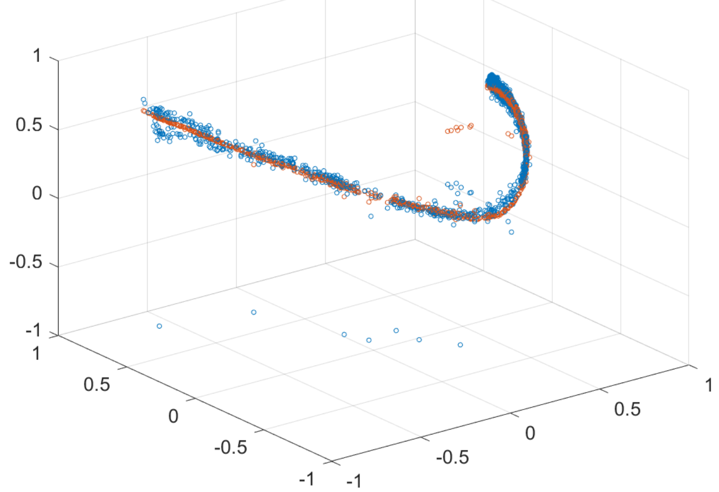
Disadvantages:

- The curves and therefore the degree of the polynomial we are fitting maybe be dataset dependent. In our case, a two degree polynomial is sufficient for identifying the clusters.

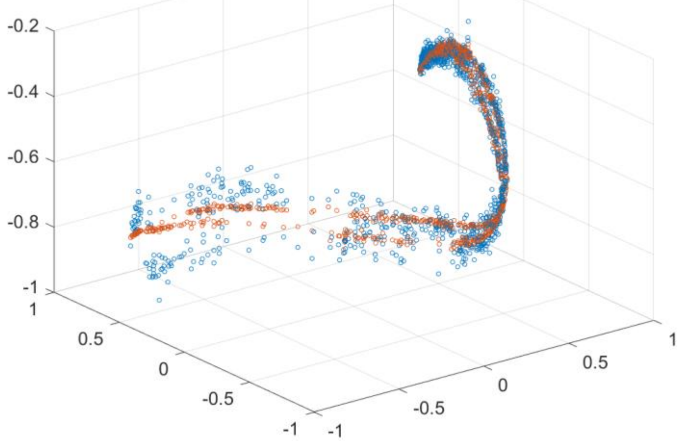
Original Clusters



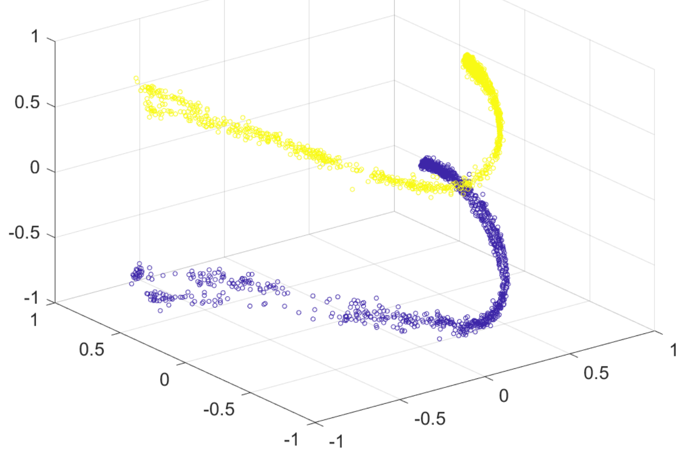
Polynomial fitted to one group
(due to RANSAC the outliers are ignored)



Polynomial fitted to second group
(due to RANSAC the outliers are ignored)



Re-classification of projections
All clusters are accurately identified



Tomographic reconstruction of symmetric objects

- The optimization problem constructed to account multiple axis of symmetry is the following – (shown below is the case for two axis of symmetry)

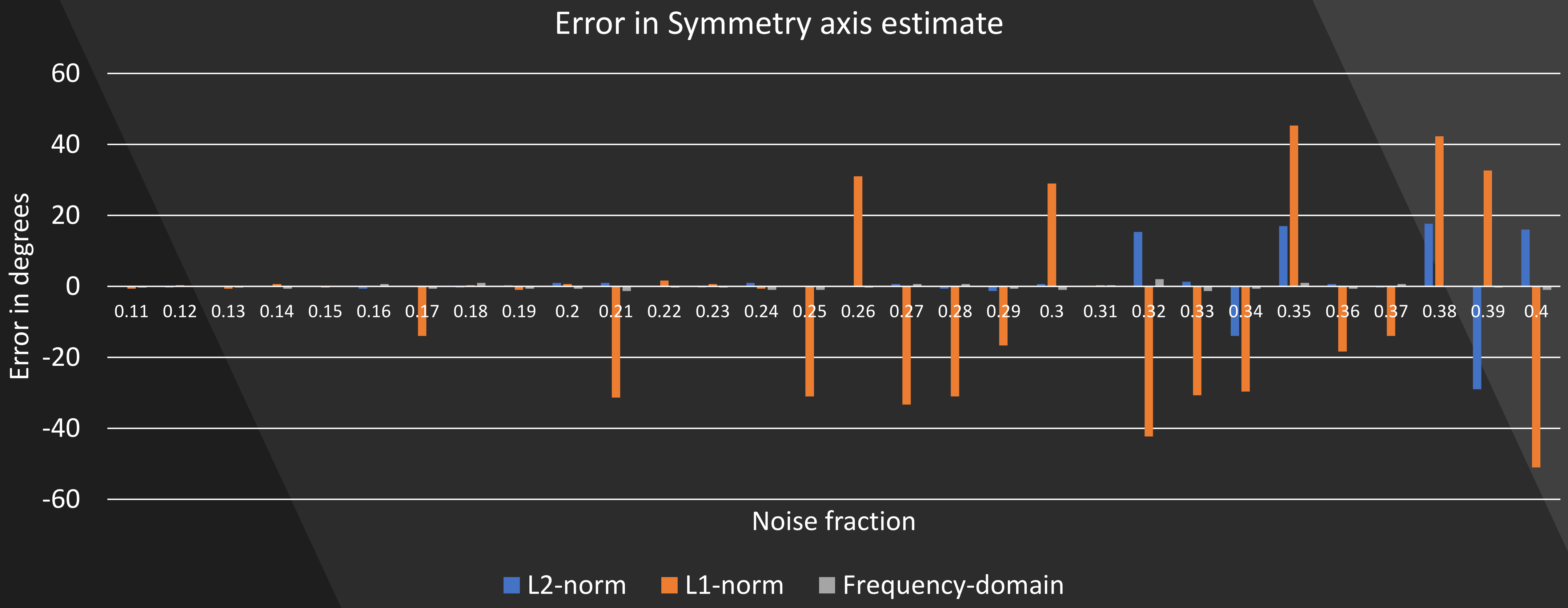
$$\sum_{i=1}^{Q} \left\| y_i - \phi_{\theta_i+\delta} x \right\|^2 + \left\| (R_\delta x)_{i,j} \;\; - (R_\delta x)_{-i,j} \right\|^2 + \left\| (R_{\delta+90} x)_{i,j} \;\; - (R_{\delta+90} x)_{-i,j} \right\|^2$$

- The metric used to detect the axis of symmetry is the following –

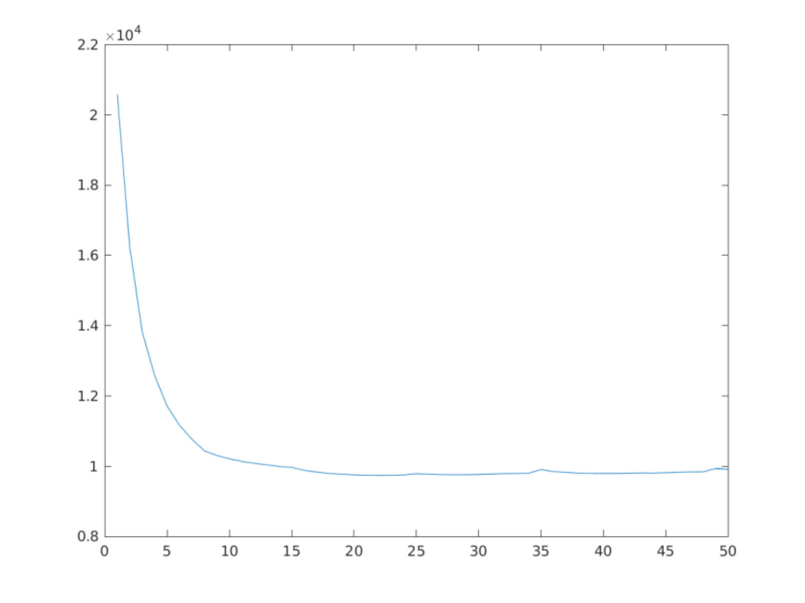
$$S\{f(x)\} = \frac{\|f_s(x)\|^2}{\|f_s(x)\|^2 + \; \|f_{as}(x)\|^2}$$

- Where $f_s(x)$ and $f_{as}(x)$ denote the symmetric and ani-symmetric parts of the function.
- The reason we choose this metric over more simpler metrics like L1-norm and L2-norm is because this metric is more robust to high noise levels. Even at noise levels upto 40%, this metric is able to accurately identify the symmetry axis, as the next graph will show.

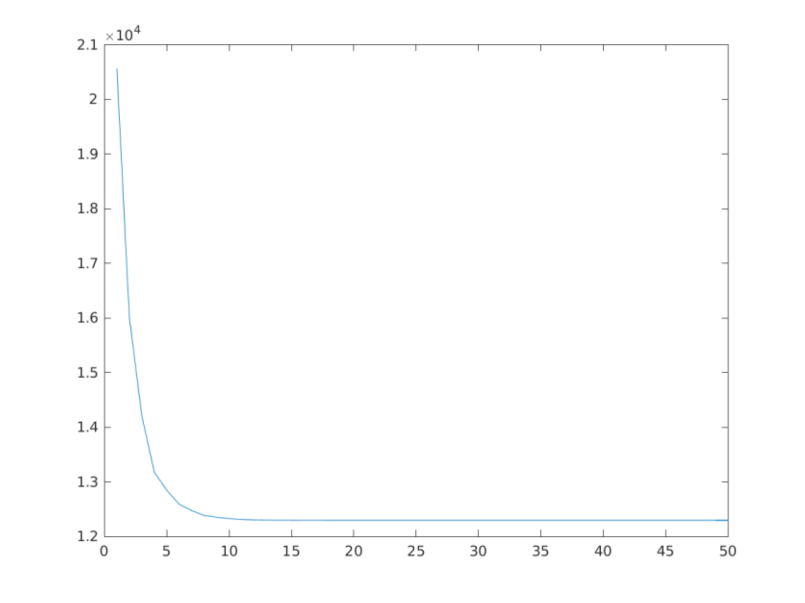
Comparison of symmetric metrics



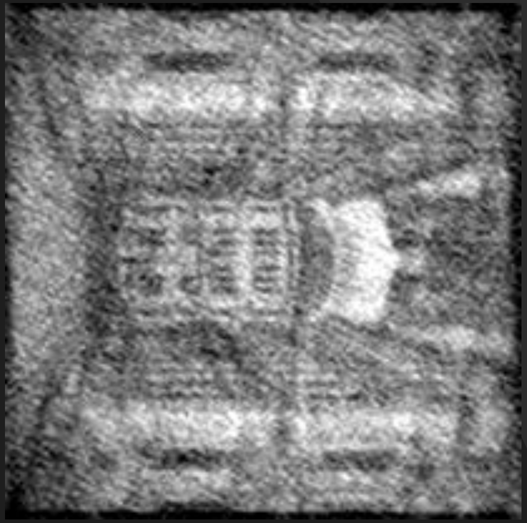
Relative reconstruction error	
Refinement without using symmetry	13.19%
Refinement using symmetry	8.55%



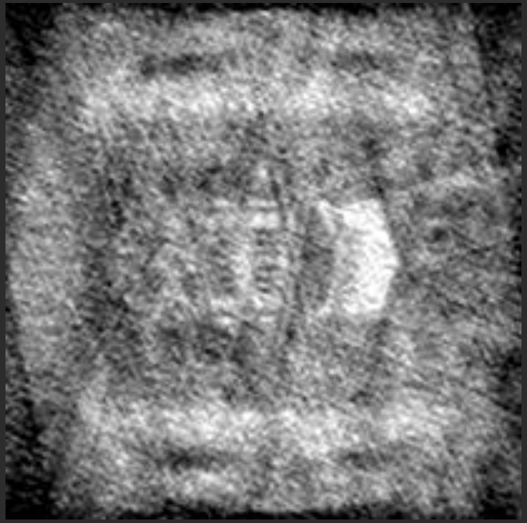
Refinement using symmetry



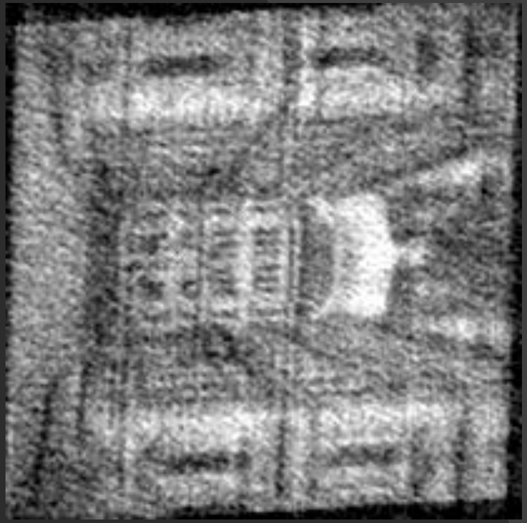
Non-symmetric convergence



Refinement without using symmetry



Estimate returned by HLCC



Symmetric convergence