Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Ultramicroscopy 108 (2008) 327–338
ultramicroscopy
www.elsevier.com/locate/ultramic

Classification of heterogeneous electron microscopic projections into homogeneous subsets

G.T. Herman*, M. Kalinowski

Department of Computer Science, Graduate Center, City University of New York, USA

Received 5 December 2006; received in revised form 30 April 2007; accepted 8 May 2007

Abstract

The co-existence of different states of a macromolecular complex in samples used by three-dimensional electron microscopy (3D-EM) constitutes a serious challenge. The single particle method applied directly to such heterogeneous sets is unable to provide useful information about the encountered conformational diversity and produces reconstructions with severely reduced resolution. One approach to solving this problem is to partition heterogeneous projection set into homogeneous components and apply existing reconstruction techniques to each of them. Due to the nature of the projection images and the high noise level present in them, this classification task is difficult. A method is presented to achieve the desired classification by using a novel image similarity measure and solving the corresponding optimization problem. Unlike the majority of competing approaches, the presented method employs unsupervised classification (it does not require any prior knowledge about the objects being classified) and does not involve a 3D reconstruction procedure. We demonstrate a fast implementation of this method, capable of classifying projection sets that originate from 3D-EM. The method's performance is evaluated on synthetically generated data sets produced by projecting 3D objects that resemble biological structures.

Keywords: Classification; Electron microscopy; Heterogeneous projections; Macromolecules; Graph cutting

1. Introduction

The analysis of macromolecular complexes and their dynamics is one of the intensively researched topics in molecular biology. The objective of this research is to understand the structure and function of molecular machines. The three-dimensional reconstruction from electron-microscopic images (3D-EM) of macromolecular complexes plays an essential role in these efforts. The single particle reconstruction method is frequently used to obtain 3D models of large macromolecular complexes. In this method, experimental projections from many (randomly oriented) specimens of the same macromolecule are treated as if they were obtained from the same specimen at different directions. The unknown projection angles are estimated under the assumption that a unique ''reference'' object (to be reconstructed) exist. Thousands of 2D projection images and their estimated angles are used as an input to the reconstruction algorithm, which returns an estimate of the 3D object from which the projection images were obtained. Despite of the very high level of noise (and other distortions) present in images produced by EM, this approach has been successfully used to obtain the 3D structural models of many biological molecules [1].

Due to the dynamic nature of molecular processes, it is quite common that a molecule has several different conformations, which are not separable prior to the process of obtaining projections. This results in heterogeneous projection sets (sets containing projections of more than one conformation). When the single particle approach is applied to such sets, its essential assumption that all specimens used in the process are identical is violated. The traditional reconstruction procedures used in 3D-EM involve a certain degree of averaging, either at the level of 2D images or of 3D objects, and so they are inherently unsuitable to deal with structural heterogeneity. The averaging across several conformations severely limits the

*Corresponding author.
E-mail address: gabortherman@yahoo.com (G.T. Herman).

0304-3991/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.ultramic.2007.05.005

achievable resolution and results in loss of the information about individual conformations, which might be essential to understanding the function of macromolecule under study.

Since the 3D visualization of molecular machines in their various conformations is essential to understanding the dynamic processes they undergo, significant efforts have been made to develop techniques to deal with heterogeneity in projection sets. Due to the inherent difficulties, the majority of the proposed methods make use of supervised classification techniques to expand the applicability of the single particle approach to heterogeneous projection sets. The expansion typically consists of estimating the projection directions and the conformations that gave rise to the projections using a set of models (or "references," and hence such approaches are referred to as multireference 3D projection alignment). Penczek et al. [2] provide a recent example of such an approach to classification. They also provide an excellent description of the problem and a survey of the literature to date; we therefore need not repeat that here. The methods based on supervised classification have a serious limitation: they are applicable only if appropriate models can be found. Bad choice of the models can lead to results reinforcing wrong assumption.

For this reason, some of the current research concentrates on unsupervised classification methods [3,4]. The performance of the cluster tracking method of Fu et al. [3] has been so far demonstrated only on a relatively small, highly oversampled region of angular space. Since cluster tracking cannot be applied to regions that are sparsely populated with the projections, this classification method needs data sets containing a large number reasonably evenly distributed projections. The method proposed by us below has no such restriction. The maximum likelihood method of Scheres et al. [4] has been demonstrated to produce good classification results. In order to solve the classification problem, this method estimates at the same time each of the 3D conformations, the angles determining each of the projections, the level of noise in the data and the level of misalignment between the projections. The computational costs of doing all these are very high: on typical desktop computers in current use, the classification time for the method of Ref. [4] is measured in months, while for the method proposed below it is measured in hours. (In both cases the waiting time for the results can be very much reduced by the use of multiple processors.)

2. Proposed approach

Our approach to the heterogeneity problem is to employ an unsupervised classification procedure to partition heterogeneous projection sets into homogeneous components. If a correct partitioning can be obtained, independent reconstructions from each of the homogeneous components will produce models of all the conformations represented in the heterogeneous set. This set of models will constitute the solution to the problem of reconstruction from a heterogeneous set. This classification-based approach is appealing, because it separates the issue of heterogeneity from that of reconstruction, allowing the use of existing reconstruction techniques without modifications. Since our approach is based on an unsupervised classification procedure, it does not suffer from many limitations of the existing methods. It does not assume any references, it does not even make any statistical assumptions about them, and so it avoids any bias that such assumptions might introduce. However, due to high level of noise present in micrographs and the nature of the projection images, the unsupervised partitioning of heterogeneous projection sets produced by EM into homogeneous components is inherently difficult.

3. Similarity measure

The difficulty inherent in our image classification problem comes from the fact that frequently a pair of images that belong to the same class (they are 2D projections of the same conformation) are far less similar to each other than another pair of images that belong to different classes. A 2D image classification method would not consider the image pairs that are within the individual panels of Fig. 1 to be similar and would be likely to place them in different classes, even though the images of these pairs are 2D projections of the same 3D object in different directions. On the other hand, it would consider that the images in each column of Fig. 1 are similar to each other, even though they are 2D projections of different 3D objects. In order to overcome this difficulty we propose a new image similarity measure, specifically designed to deal with 2D projections of 3D objects. This measure utilizes

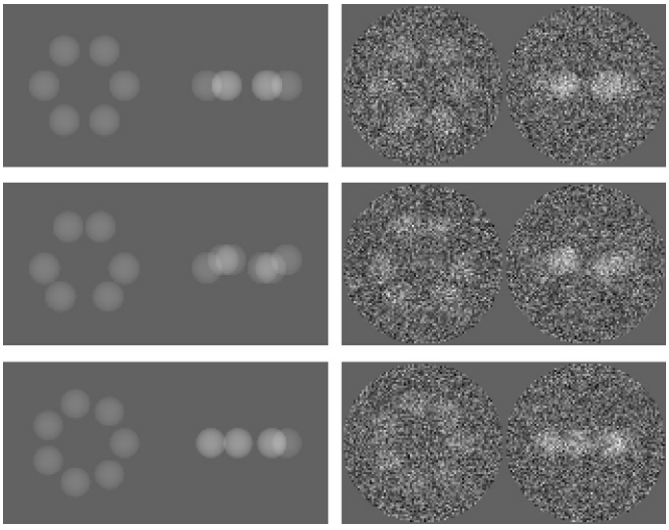

Fig. 1. 3D objects used in the experiments (top and side views). Left column: two noiseless 2D projections of object S6 (top), S6x (middle), and S7 (bottom). Right column: corresponding noisy projections with SNR = 0.1. (All images use the same mapping from projection values into gray values; the two extreme projection values in the noisy images correspond to black and white.)

a property of images that are 2D projections of the same 3D object and, consequently, it is well suited to our classification problem.

Our projection image similarity measure is based on the following mathematical argument. The value at a point in a perfect (i.e., noiseless) projection image is the line integral of the 3D object to be reconstructed along a line that goes through that point and is orthogonal to the plane of the projection image. An immediate consequence of this is that if we take any line in the projection image and we integrate the projection values along that line, the resulting line integral will have the same value as the planar integral of the original 3D object over a plane that contains the given line in the projection plane and is orthogonal to the projection plane.

Consider now two perfect projection images x and y (in different directions) of the same 3D object (Fig. 2). The planes (p and q) of these projections intersect in a line (cl). This line occurs in both of the projection planes, and it is therefore referred to as the *common line*. Take any point (a) on the common line and consider the two lines ($l_{a,p}$ in p and $l_{a,q}$ in q) that are perpendicular to the common line and include this point. These two lines lie in the same plane (P_a, which is perpendicular to the common line cl and goes through the selected point a). It follows therefore that the line integrals that are obtained by integrating in the projection images (x and y) along the lines perpendicular to the common line (cl) going through the selected point (a) have the same value (namely, that of the planar integral of the original 3D object over the plane P_a).

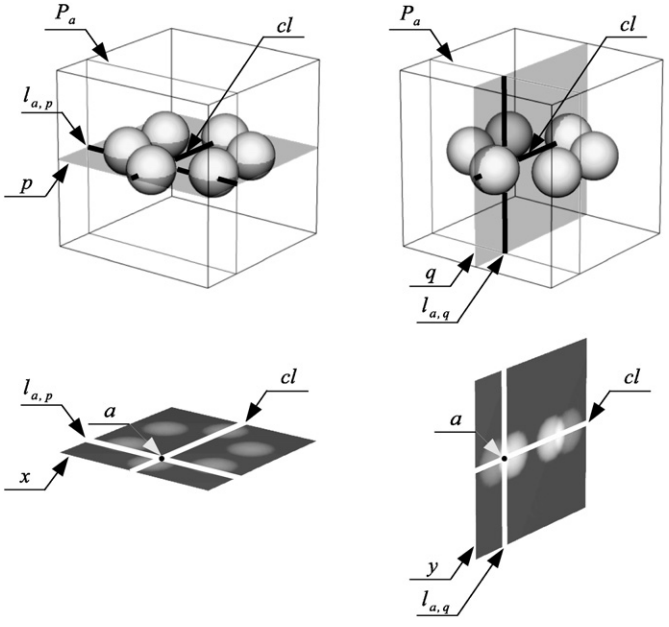

Fig. 2. Two projections x and y of object S6. The line integrals in projection planes (p and q) along the indicated lines ($l_{a,p}$ and $l_{a,q}$) perpendicular to the common line (cl) are both equal to the planar integral over the indicated plane of integration (P_a). (P_a—plane of integration; $l_{a,p}$, $l_{a,q}$—lines perpendicular to the common line; cl—common line; x, y—projections onto projection planes; p, q—projection planes; a—point on the common line).

An important (and well-known [5]) consequence of this is the following property of any two perfect projection images of the same 3D object. Let us assume that we can identify in the projection planes (p and q) of the two projections the points (call them o_p and o_q, respectively) onto which the origin of the assumed coordinate system of the 3D Euclidean space project. Here we will not discuss further how to go about finding these points in case of noisy projections, since such a task needs to be performed for 3D-EM in any case (in order to bring the various projection images into a common coordinate system) and is much discussed elsewhere in the literature. (In Sections 6 and 8 we report on the results of some classification experiments for projection images in which the assumed locations of o_p and o_q are not accurate.) Assuming that we have the desired o_p and o_q, we can translate the property described in the previous paragraph into the following: there is a line (call it l_p) in p going through o_p and a line (call it l_q) in q going through o_q such that, for any real number d, the value of the line integral of the projection x onto p along the line perpendicular to l_p at a distance d from o_p is the same as the value of the line integral of the projection y onto q along the line perpendicular to l_q at the distance d from o_q.

In practice, we have to deal with the discrete and noisy nature of our data (which implies that the planar integrals of the 3D object can be determined from the projection images only approximately) and computational reality does not allow us to look at all lines in the projection planes. Let L be the number of evenly distributed lines at which we will look in each projection plane p, we index them by l, $1 \leqslant l \leqslant L$. Each of these lines go through o_p. On each of them we pick C points (these points are picked at matching distances from o_p in all projection planes p and for all lines l). For each projection image x and for each line l we define a C-dimensional vector X_l whose cth component (for $1 \leqslant c \leqslant C$) is the estimated line integral in the projection image along the line perpendicular to l going through the cth point. If errors due to noise and discretization are ignored, then (according to the property described above) two projection images x and y of the same 3D object must have identical vectors X_l and Y_m for some pair of indexes l and m. This can also happen if x and y are projection images of different 3D objects, but only under very special circumstances. In reality, due to discretization error and noise, there is practically no pair of indexes l and m for which vectors X_l and Y_m are identical. However, there is an increased probability of finding two "similar" vectors X_l and Y_m, if the projections x and y came from the same object. Let us assume that 'similarity' of vectors can be measured by a function D that returns 0 given a pair of identical vectors and a positive value indicative of the differences between the vectors otherwise. We define the distance of any two projection images x and y as

$$d(x,y) = \min_{1 \leqslant l,m \leqslant L} D(X_l, Y_m). \quad (1)$$

For our current work we have chosen $D(X_l, Y_m) = \|X_l - Y_m\|^2$ (the squared 2-norm of the difference).

4. Projection image classification as optimization problem

The high level of noise present in the projections sets obtained by EM causes that even when an appropriate similarity measure (as defined by Eq. (1)) is used, the task of identifying homogeneous components in such sets is difficult. The value of the similarity measure between two projections produced from the same conformation is only statistically smaller than the value for two projections of different conformations. In fact when similarity measures are calculated for all the pairs of projections in a realistic heterogeneous projection set, the range of values for pairs originating from the same conformation is practically the same as the one for the pairs originating form different conformations. (For example, using the data set generated for the experiment of Table 5, the first range is from 0.84×10^6 to 2.21×10^6 with mean 1.57×10^6 and standard deviation 0.12×10^6, while the second range is from 0.79×10^6 to 2.44×10^6 with mean 1.63×10^6 and standard deviation 0.13×10^6. The histograms of distances in these sets are shown in Fig. 3.) In such circumstances, only approaches that simultaneously consider many (possibly all) projections have the potential of finding the correct partitioning.

In order to obtain reliable classification of noisy projection images (such as those produced by EM), we propose the reformulation of the original classification problem as the following optimization problem. Let V denote the heterogeneous projection set. For any positive integer K, a *K-partition* A of V is a set $\{A_1, \ldots, A_K\}$ of K nonempty subsets of V such that the union of these subsets is the whole of V and no two subsets have any element in common. Our reformulated problem is:

GIVEN a set V of 2D projections and a positive integer K,
FIND a K-partition $A = \{A_1, \ldots, A_K\}$ of V,
SUCH THAT

$$\sum_{k=1}^{K} \sum_{x,y \in A_k} d(x, y) \tag{2}$$

is as small as possible.

This problem can also be restated in the context of graph theory [6]. The projections in V are represented by nodes of a complete weighted graph G, the weight of the edge between nodes x and y is the distance $d(x, y)$, defined by Eq. (1). In such a graph, the edges between the nodes representing projections of the same object are more likely to have lower weights. The problem of separating the homogeneous subsets of a heterogeneous projection sets becomes a graph cutting problem, in which the objective is to find a separation of the graph G into K complete subgraphs $G_1, ..., G_K$ such that the sum of all edge weights in the subgraphs $G_1, ..., G_K$ is minimal. This problem is known as Max k-Cut [7], and in case $K = 2$ it is equivalent to the maximum capacity cut problem [8]. Fig. 4 shows a small (10 images) instance of heterogeneous projection set classification problem interpreted as maximum capacity cut problem of the corresponding complete graph.

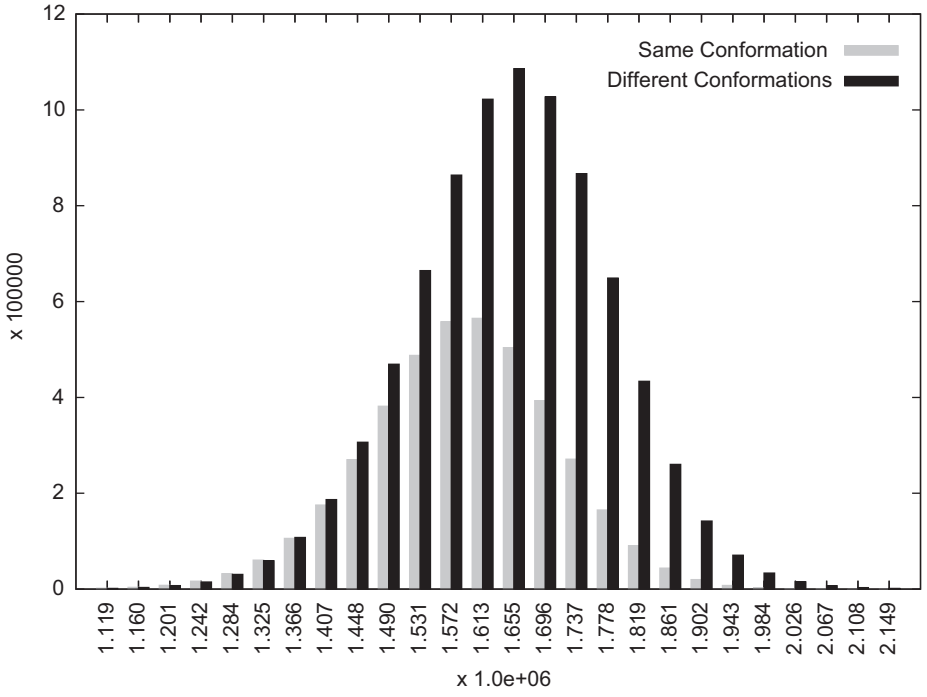

Fig. 3. Histograms of distances between pairs of projection images in a heterogeneous set for the pairs originating from the same and from different conformations.

The translation of the biological classification problem into our proposed optimization problem is appropriate for the situations in which the projection images being classified contain significant amount of noise, because the classification of the individual images is not made independently. In fact, the classification of each projection image is determined by the optimal arrangement of all the images and therefore it is much less likely to be influenced by noise. However, according to theory, finding an optimal (or even approximately optimal) solution to a large instance of the Max k-Cut or the maximum capacity cut problem is extremely computationally expensive. Both these problems are proved to be NP-complete (which means that their larger instances are computationally intractable [9]), and so for larger graphs one cannot in general expect that an algorithm will find an approximate minimizer of the expression of Eq. (2) in an acceptable computer time. Nevertheless, we provide below an algorithm that does exactly that for the instances of the Max k-Cut and the maximum capacity cut problems that come from 3D-EM.

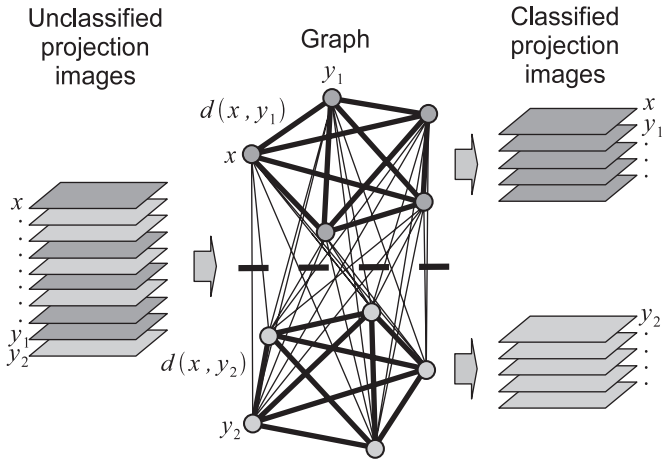

Fig. 4. Classification by graph cutting. Images represented by nodes of each graph component belong to the same class. The thick edges connect the graph nodes that represent pairs of similar images (those for which the value of Eq. (1) is low). The thin edges connect the graph nodes that represent pairs of dissimilar images (those for which the value of Eq. (1) is higher).

5. Method

Our method combines the proposed similarity measure and the optimization based approach described above (see Fig. 5).

Prior to further processing, all projection images are masked with a circular mask using the following

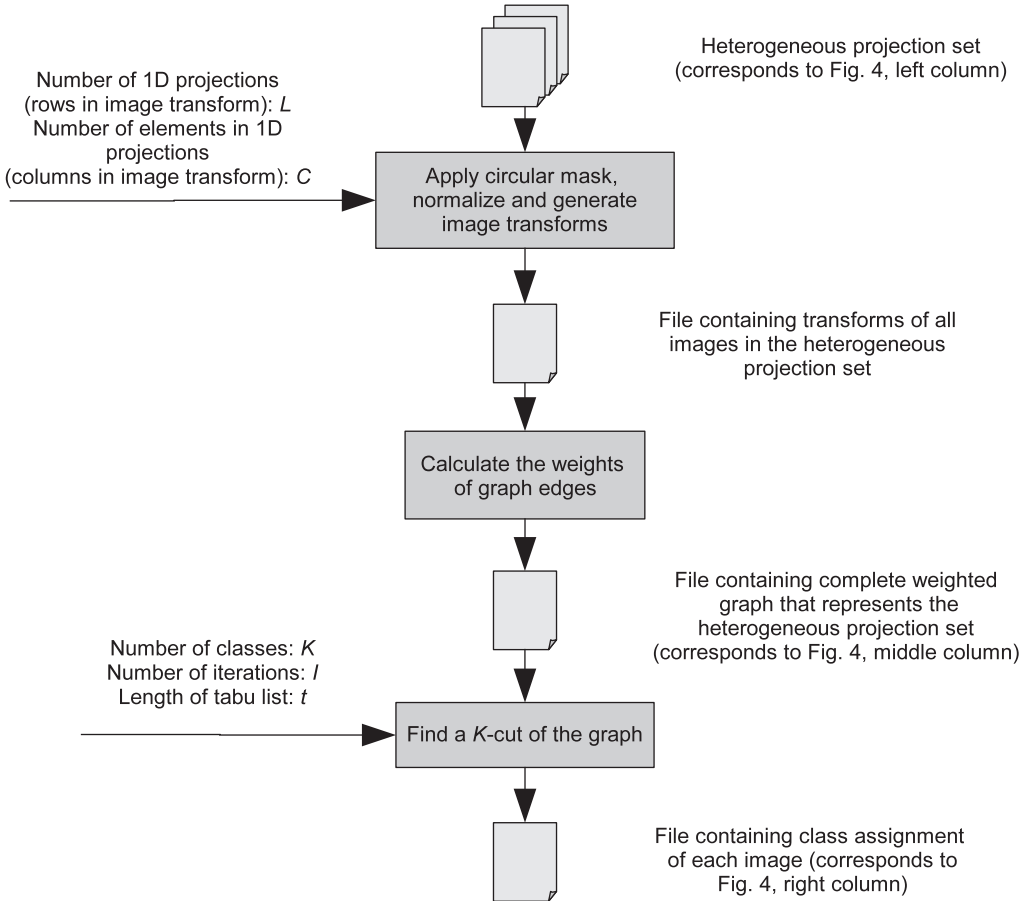

Fig. 5. Classification procedure (data-flow).

procedure. The radius of the masking circle is selected to be large enough to enclose all the projections of the supports of the 3D objects. The center of the masking circle is placed in the center of the image. The pixel values outside the masking circle are set to zero. In addition, the pixel values of each image are normalized by subtracting a constant value from the pixel values within the masking circle so that their sum after subtraction is zero. Such a normalization is justified by the nature of the contrast transfer function of an electron microscope (the information about the sum of densities is essentially lost [1]) and the facts that it does no harm in the noiseless case but eliminates the average of the noise in the noisy case.

These preprocessed images are used to construct a complete weighted graph. Each node of this graph corresponds to a single 2D projection in the noisy heterogeneous set. The weight of each edge in the graph is equal to the value of the similarity measure between the images represented by the nodes it connects. In practice, to reduce required computation, this is done in two steps. First we produce L C-dimensional vectors X_l from each preprocessed 2D projection image x, by projecting the image x in L evenly spaced directions. These vectors are stored and repeatedly used in the second step, in which we compute the weight of each edge in the graph according to Eq. (1). This is done by calculating the value of function $D(X_l, Y_m)$ for all pairs of vectors X_l and Y_m, that were generated (in the first step) from the images x and y represented by the two ends of the edge.

Despite the fact that according to theory finding even an approximate solution to large instances of the Max k-Cut problem is extremely computationally expensive [7], we were able to construct an algorithm that finds good (from the perspective of our classification) approximate solutions to the instances of Max k-Cut problem that originate from heterogeneous projection sets (even those that contain thousands projections). Our graph cutting algorithm is based on *tabu search*, a standard method of combinatorial optimization [10]. The idea is to make changes to the current guess at the optimal partition to minimize the desired functional as much as possible, but to avoid getting trapped at a local (rather than the global) minimum use a (tabu) list of recently switched elements that are (generally) not allowed to be switched back again. A simplified flowchart of the algorithm is provided by Fig. 6. The exact operation is described by the following pseudo-code:

Parameters:

K: number of classes
I: number of iterations
t: length of tabu list

Input:

G : complete weighted graph with nodes V and edges $E = \{(v_1, v_2, w) | v_1, v_2 \in V, w \text{ is a real number}\}$

Internal objects and functions:

M: mapping vector that assigns to each element $v \in V$ a label $M(v) \in \{1, \ldots, K\}$ (this label determines the partition of V to which v belongs)

T: tabu list of length t that contains pairs $(v, T(v))$, where $v \in V$ and $T(v)$ is a real number

$S(G, M)$: objective function (to minimize), $S(G, M)$ returns the sum of the weights w of edges (v_1, v_2, w) in graph G such that $M(v_1) = M(v_2)$

$S_{\text{new}}(G, M, v, k)$: function that returns the sum of the weights w of edges (v_1, v_2, w) in graph G such that $M_{\text{new}}(v_1) = M_{\text{new}}(v_2)$,

where $M_{\text{new}}(x) = \begin{cases} M(x) & \text{for } x \neq v \\ k & \text{otherwise} \end{cases}$

Output:

M_{opt} : best mapping vector found by the algorithm

Algorithm (*single run*):

FOR ALL $v \in V$
  $M(v) = randomvalue\ from\ \{1, \ldots, K\}$
$M_{\text{opt}} = M$
FOR $i = 1$ to I
  find a pair $v \in V$ and $k \in \{1, \ldots, K\}$ such that
    $k \neq M(v)$
    and
      v is not on tabu list
      or
      $T(v) > S_{\text{new}}(G, M, v, k)$
  for which $S_{\text{new}}(G, M, v, k)$ is minimal
  if v is on tabu list then
    $T(v) = \min\{S_{\text{new}}(G, M, v, k), S(G, M)\}$
  else
    if the tabu list reached its maximum size t then
    remove last item
    add pair $(v, \min\{S_{\text{new}}(G, M, v, k), S(G, M)\})$ to the front of the tabu list
    $M(v) = k$
  if the value of $S(G, M)$ is smaller than any seen so far then $M_{\text{opt}} = M$

Both the functions $S(G, M)$ and $S_{\text{new}}(G, M, v, k)$ can be implemented very efficiently, by maintaining edge sum tables. The complete contents of these tables need to be calculated only once (during initialization) and only local updates are necessary after each iteration.

Since the algorithm finds only an approximation (that is dependent on the initial random classification of nodes), it is advisable to run it repeatedly to ensure that we find a good (or, at least, a robust) approximation to

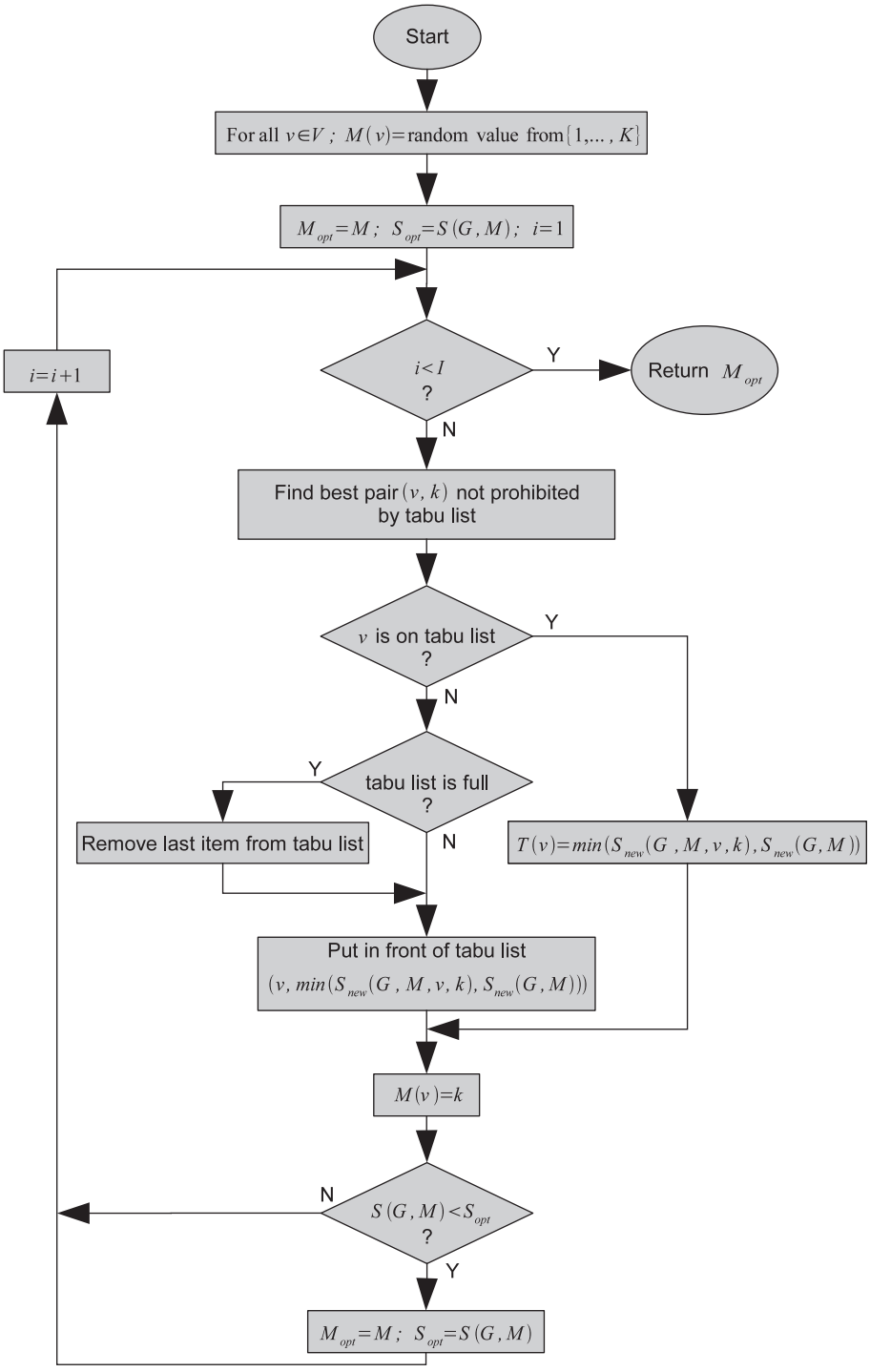

Fig. 6. Graph cutting algorithm—single run. Input: graph $G = (V, E)$. Parameter: number of iterations I. Returns: optimal mapping vector M_{opt}.

the optimal classification. Since the algorithm is very inexpensive (as compared to the cost of producing the graph that is to be cut), this does not seriously increase the overall cost.

Our algorithm requires that the number of classes be provided. For a heterogeneous projection set with unknown number of represented objects, one can run (inexpensively) the algorithm for different values of K.

6. Experiments

We carried out a series of experiments to evaluate the proposed method. All data sets used in these experiments were generated synthetically. We carefully selected their parameters to make them as realistic as possible. We used three 3D objects, referred to as S6, S6x, S7 in Fig. 1, which closely resemble objects previously utilized in 3D-EM

literature [11,12]. The level of noise we inserted into the projections is not atypical for real electron micrographs of biological macromolecules: signal-to-noise ratio (SNR) 0.1 (as defined by Frank [1], p. 121). As in real heterogeneous projection sets, in some of our test sets conformations are unevenly represented (i.e., the majority of the projections present in the set comes from one conformation).

6.1. Projection data generation

The following method was used to generate heterogeneous projection sets for the initial experiments. For each of the three 3D objects (S6, S6x, S7) we generated 2562 noiseless 2D projections (81×81 pixels) with evenly distributed projection angles. This produced three noiseless homogeneous projection sets (see Fig. 1, left column). The heterogeneous sets containing projections of two conformations were generated by randomly selecting (with possible repetitions) N_1 projections from the homogeneous set for the first object and then randomly selecting (with possible repetitions) N_2 projections from the homogeneous set of the second. The total number of projections in resulting heterogeneous set is $N = N_1 + N_2$. For each pair of objects (S6–S7, S6x–S7, S6–S6x), we used three different ratios (50:50, 35:65, 20:80) between the values N_1 and N_2 to produce heterogeneous projection sets with uneven representation of the objects. In our experiments we set $N = 5000$ and $N_1 = 2500$, $N_2 = 2500$ for the 50:50 sets, $N_1 = 1750$, $N_2 = 3250$ for the 35:65 sets, and $N_1 = 1000$, $N_2 = 4000$ for the 20:80 sets. In order to obtain statistically reliable results, for each pair of objects (S6–S7, S6x–S7, S6–S6x) at each object representation ratio (50:50, 35:65, 20:80), the random selection was executed five times. This produced 45 noiseless heterogeneous projection sets, each containing projections of two 3D objects, in nine groups (S6–S7_50:50, S6x–S7_50:50, S6–S6x_50:50, S6–S7_35:65, S6x–S7_35:65, S6–S6x_35:65, S6–S7_20:80, S6x–S7_20:80, S6–S6x_20:80). The data sets in each group were produced from the same pair of objects and had the same representation ratio.

A similar procedure was used to produce heterogeneous sets containing 2D projections of all three 3D object. In the experiments reported here, we used $N = 5000$ and $N_1 = 1667$, $N_2 = 1667$, $N_3 = 1666$. As in the case of sets containing two 3D object, the random selection was executed five times. This resulted in five sets (referred to as group S6–S6x–S7_33:33:33), each containing 2D projections of three 3D objects (with equal object representation).

The method of generating heterogeneous projection sets described above guarantees that all the projections within a projection set are perfectly aligned (the geometric center of each configuration is projected exactly to the center of the projection image). In practice, a perfect alignment of very noisy EM projections cannot be achieved. Since the misalignment of the projection images has a potentially negative impact on the performance of proposed method, an additional 50 heterogeneous projection sets, designed to test the impact of projection misalignment, where generated by the following method.

First, for each of the heterogeneous projection sets 5000 projection angles where randomly (with potential repetitions) selected in such a way that all the projection angles were equally likely. Each of these angles was used to project one of the objects to be represented in the projection set being constructed. In 50:50 case, the first 2500 angels were used to project the first object and the remaining 2500 angles were used to project the second object (in the case of the 20:80 and 35:65 data sets, appropriately adjusted numbers were used). However, before producing a projection image, the center of the 3D object was shifted in a randomly selected direction parallel to the projection plane by a distance selected from the Gaussian distribution with mean zero and standard deviation equal to the 1/81 of the diameter of the masking circle. For each pair of objects (S6–S7, S6x–S7, S6–S6x) at each object representation ratio (50:50, 35:65, 20:80), the random selection of projection angles and shifts was executed five times. This produced 45 noiseless heterogeneous projection sets, each containing misaligned projections of two 3D objects, in nine groups (S6–S7_Sh_50:50, S6x–S7_Sh_50:50, S6–S6x_Sh_50:50, S6–S7_Sh_35:65, S6x–S7_Sh_35:65, S6–S6x_Sh_35:65, S6–S7_Sh_20:80, S6x–S7_Sh_20:80, S6–S6x_Sh_20:80). Similar procedure was used to produce five heterogeneous sets (data set group S6–S6x–S7_Sh_33:33:33) containing misaligned 2D projections of all three 3D objects.

Prior to applying the noise insertion procedure to the noiseless heterogeneous projection sets, we randomly reordered them to avoid any influence of the order in which images were generated on the results of experiments. In our experiments we modeled noise by adding to each pixel of a projection the value of a random variable selected from a Gaussian distribution with zero mean and appropriately selected variance. Following the definition of SNR in Ref. [1] on p. 121, we calculated the signal variance σ_S^2 as the intensity variance of pixels (within the masking circle) in all the noiseless 2D projections and the noise variance as $\sigma_N^2 = \sigma_S^2/\mathrm{SNR}$.

6.2. Method parameters

We used code from SNARK05 [13] to produce, for each of the masked 2D images, 240 1D projections at 1.5° angular increments (240 is the value of L in Eq. (1)) with line integrals calculated for 81 equally spaced lines for each 1D projection (81 is the value of C in the discussion above). In all the reported experiments the classification algorithm was executed 10 times, each time starting with different randomly selected initial assignment. In all these runs the length of the tabu list (t) was set to 3000 and the algorithm was terminated after executing 10,000 iterations.

6.3. Performance evaluation

To evaluate the performance of the proposed method we use the figure of merit called classification purity [14] in %,

Table 1
Examples of the results from two-class classification experiments with conformation representation ratios 50:50, 35:65, and 20:80

Representation ratio	Projections of object	No. of projections assigned to	
		Class 1	Class 2
50:50	S6x	33	2467
	S7	2499	1
35:65	S6x	0	1750
	S7	2559	691
20:80	S6x	8	992
	S7	2535	1465

Table 2
Mean classification purity when the number of classes is appropriate for the conformation representation ratio (based on five data sets for all nine groups)

Data sets	Mean classification purity %		
	50:50→2	35:65→3	20:80→5
S6–S7	98.73 ± 0.06	95.97 ± 0.11	95.66 ± 0.12
S6x–S7	99.38 ± 0.02	97.77 ± 0.01	98.91 ± 0.03
S6–S6x	98.71 ± 0.05	95.80 ± 0.07	96.09 ± 0.52

which is calculated as follows. We create an array whose rows correspond to the conformations and whose columns correspond to the classes produced by our algorithm; an entry in the array is the number of projection images from the corresponding conformation that are put by our algorithm into the corresponding class. For example, Table 1 exhibits three such arrays. Ideally, all elements of a class should come from the same conformation. We therefore define the *classification purity in* % as: 100 times the sum over columns of the maximum of the entries in the column, divided by the sum of all the entries in the table. For the three arrays in Table 1, the classification purities are 99.32%, 86.18%, and 80.00%, respectively.

Each experiment we conducted involved one group of data sets. For each group of data sets used in the experiment, we report the mean classification purity in % (m) and the standard error of the mean (e) using the notation $m \pm e$; see, for example, Table 2.

7. Results

In our first set of experiments two conformations were evenly represented in the projection sets. For these experiment we used data set groups S6–S7_50:50, S6x–S7_50:50, S6–S6x_50:50 and configured our classification algorithm to identify two distinct classes. Table 1 (row: 50:50) provides a typical result. The results of these experiments are summarized in Table 2 (column: 50:50→2). For all data sets in these experiments the classification purity was 98.5% or higher.

Since there is no reason to assume that in real heterogeneous projection sets objects are evenly represented, we designed a second set of experiments to evaluate the performance of our method when it is applied to sets with asymmetric representation of conformations. In these experiments we used two types of sets, with representation ratios 35:65 (data set groups: S6–S7_35:65, S6x–S7_35:65, S6–S6x_35:65) and 20:80 (data set groups: S6–S7_20:80, S6x–S7_20:80, S6–S6x_20:80). We set the classification algorithm to identify two distinct classes. A typical result for representation ratio 35:65 is provided in Table 1 (row: 35:65), a corresponding example for ratio 20:80 is provided in Table 1 (row: 20:80). These results clearly indicate that our method, when applied to realistic (from the EM perspective) projection sets with asymmetric representation, has a bias toward creating evenly sized classes. This has the effect on the produced classification that one of the classes contains almost exclusively projections of the conformation from which the majority of the projections in heterogeneous set was obtained and the other contains a mixture of the projections of both conformations. This classification successfully isolates a sufficiently large homogeneous projection set for one of the conformations to allow its reconstruction. The object reconstructed from the other class will likely exhibit artifacts indicating that it was reconstructed from a heterogeneous set. One of our approaches to correctly identifying the conformation represented by the minority of the projections is to partition the original projection set into a larger number of classes.

In the next two sets of experiments we evaluated the performance of our classification algorithm when setting the number of classes to three for sets with representation ratio 35:65 and to five for sets with representation ratio 20:80. Typical results in these experiments are provided in Tables 3 and 4. The classification purities for all data set groups in these two experiments are summarized in Table 2 (columns: 35:65→3 and 20:80→5). Despite of the asymmetric representation, our method (when using the appropriate number of classes) was able to produce a

Table 3
Example of the results from the three-class classification experiment with conformation representation ratio 35:65

Projections of object	No. of projections assigned to		
	Class 1	Class 2	Class 3
S6x	18	95	1637
S7	1674	1575	1

Table 4
Example of the results from the five-class classification experiment with conformation representation ratio 20:80

Projections of object	No. of projections assigned to				
	Class 1	Class 2	Class 3	Class 4	Class 5
S6x	958	1	3	35	3
S7	9	1015	1004	961	1011

classifications with classification purity 95.36% or higher, which allows for reconstruction of all the conformations represented in the heterogeneous projection set. Of course, in case of a real heterogeneous projection set, the correct number of classes is not known. However, it can be determined experimentally by using our classification procedure to produce several different (2-, 3-, 4-, 5-, 6-fold) partitions of the heterogeneous projection set, producing 3D reconstructions from each of them and finding one for which reconstructed objects do not exhibit artifacts indicating that averaging over multiple conformations has occurred.

We also tested the performance of our classification procedure when applied to a heterogeneous projection set with even representation of three conformations. We configured our classification procedure to produce three classes. A typical result of this experiment is provided in Table 5. The mean classification purity based on five data sets was 98.3% with the standard error of the mean 0.11%.

The corresponding results of our experiments with the projection sets containing misaligned projection images of two objects (data set groups: S6–S7_Sh_50:50, S6x–S7_Sh_50:50, S6–S6x_Sh_50:50, S6–S7_Sh_35:65, S6x–S7_Sh_35:65, S6–S6x_Sh_35:65, S6–S7_Sh_20:80, S6x–S7_Sh_20:80, S6–S6x_Sh_20:80) are summarized in Table 6. The mean classification purity based on five data sets containing misaligned projections of three objects (data set group S6–S6x–S7_Sh_33:33:33) was 96.1% with the standard error of the mean 0.08%.

Each of the experiments required approximately 12 h of computation per data set on an Intel(R) Xeon(TM) CPU running at 1.70 GHz and approximately 8 h of computation per data set on an AMD Athlon(TM) 64 Processor 3200+ when code with SSE optimizations was used. Nearly all of this time was spent on calculating similarity measures for all pairs of projections (i.e., constructing the graph [15]). The graph cutting algorithm required only approximately 2 min of computation per data set. In all experiments the basic algorithm was executed 10 times for each data set with the final M_{opt} chosen as the one for which $S(G, M_{\text{opt}})$ is the smallest among the 10 outputs. This required approximately 20 min in total.

Table 5
Example of the results from the three-class classification experiment with three equally represented conformations

Projections of object	No. of projections assigned to		
	Class 1	Class 2	Class 3
S6	24	1637	6
S6x	7	29	1631
S7	1654	12	0

Table 6
Mean classification purity for misaligned projection sets when the number of classes is appropriate for the conformation representation ratio (based on five data sets for all nine groups)

Data sets	Mean classification purity %		
	50:50→2	35:65→3	20:80→5
S6–S7	88.52 ± 7.80	93.11 ± 0.21	92.92 ± 0.14
S6x–S7	98.59 ± 0.09	97.34 ± 0.05	98.08 ± 0.13
S6–S6x	97.76 ± 0.09	90.34 ± 0.43	91.71 ± 0.17

8. Discussion

The experiments involving data set groups S6–S7_50:50, S6x–S7_50:50, S6–S6x_50:50, S6–S6x–S7_33:33:33, S6–S7_Sh_50:50, S6x–S7_Sh_50:50, S6–S6x_Sh_50:50 and S6–S6x–S7_Sh_33:33:33 clearly demonstrate that the proposed method produces very good results when applied to heterogeneous data sets with even conformation representation.

The bias of the method toward even splits (solutions in which all classes contain approximately the same number of projections) observed in experiments with data set groups S6–S7_35:65, S6x–S7_35:65, S6–S6x_35:65, S6–S7_20:80, S6x–S7_20:80, S6–S6x_20:80 when number of classes was set to 2, is caused by the overlap in the ranges of the similarity measure described in Section 4 (see Fig. 3). Since all the edges in the graph have approximately the same weight, the cut that removes largest number of edges tends to minimize the sum of remaining edges. The largest number of edges is removed when the subgraphs produced by the cut have the same number of nodes, therefore the method tends to produce classes with similar size.

Our attempts to remove the bias toward even splits by modifying the objective function resulted only in limited success. In the process of developing our method, we tested many different objective functions; among them:

$$\sum_{k=1}^{K} \frac{1}{|A_k| - 1} \sum_{x,y \in A_k} d(x, y). \tag{3}$$

This function is designed to minimize the sum of the weight means of internal edges attached to each node. The use of mean in this function reduces the impact of uneven splits on its value. By treating the nodes separately and summing over their individual means, it also reinforces the fact that, when nodes are correctly classified, the mean weight of the internal edges for each of them should be as small as possible. This function reduces the bias toward even splits. Its use allowed us to achieve good classifications for all data set groups (S6–S7_35:65, S6x–S7_35:65, S6–S6x_35:65) with representation ratio 35:65 and for a single data set group (S6x–S7_20:80) with representation ratio 20:80. However, for all data set in groups S6–S7_20:80, S6–S6x_20:80 even splits resulted in the lowest value of the objective function of Eq. (3). The advantage of the function of Eq. (2) is that its bias is known and is the same for any representation ratio. Therefore, in situations where the representation ratio of the heterogeneous data set is

unknown (as in 3D-EM), the procedure for classifying that uses a larger number of classes and the function of Eq. (2) and merges those that came from the same object may well be preferable to using fewer classes and the function of Eq. (3).

To overcome the computational intractability of the NP-complete Max k-Cut problem, we relied on the fact that the graphs that we need to cut are not arbitrary but are derived from a physical process (EM). Such graphs have some special properties (for example, as it can be seen in Fig. 3, the edge weights in these graphs fall into a relatively small range and within that range they tend to center close to some fixed value). Due to such properties, the proposed algorithm is much faster for the application at hand than the general-purpose algorithms; for example, it needs only minutes for the 5000 node problem. (However, the performance of our algorithm when applied to an arbitrary graph is likely to be poor.) As a comparison, we evaluated the DSDP algorithm [16] for finding the maximum capacity cut (partitioning into two classes) of a graph. The 2000 node instance of the problem took over 50 h to solve and, based on our experience that doubling the number of nodes increases computer time by over a factor of 10, we estimate that solving the 5000 node instance by DSDP would take approximately 1 month. Our algorithm provides an acceptable solution in approximately 2 min when applied to such a large instance, provided that it arises from the problem of classification of heterogeneous 2D projections.

Since the combinatorial optimization required can be performed very fast, the initial graph construction task is the only time-consuming step in the proposed method; our proposed way of doing this is described in Ref. [15]. The computation of distances for all pairs of projections (the weights of the edges in the complete graph) is computationally expensive because of the large number of values to be computed ($N(N-1)/2$ for a graph of N nodes). However, the computations for given pairs are completely independent of each other and can be easily distributed over several computers. Even when a single computer is used, the time required (approximately 10 h for 5000 projections) is relatively small when compared to other tasks involved in the 3D-EM process.

It is important to notice that once the graph is constructed for given a data set, it can be used in many classification runs. The speed of the proposed graph cutting algorithm allows us to experiment with different number of classes to find the best classification and also to repeat the algorithm with a number of random starting partitions in order to find a good approximation to the optimum.

The results of experiments with misaligned projection sets demonstrate that our method is to some extent robust when applied to such data sets. With the exception of the one data set, the classification purity was above 89%. This resulted in high (larger than 88%) mean classification purities (reported in Table 6) for all data set groups. However, the very low value (53.66%) of classification purity in the single case for which the method failed (reflected in Table 6 by lowest mean classification purity and highest standard error of the mean for corresponding data set group) clearly indicates that the issue of misalignment cannot be ignored in practice.

There are several ways of dealing with projection image misalignment that should be considered and evaluated. One of the possible approaches is to prealign the projection images before they are used in the classification procedure. Such prealignment can be based on the use of the centers of the mass of projection images. Due to high level of noise present in the projection images, it is unlikely that this approach will result in high quality alignment. However, the resulting alignment might be sufficient to classify images and then a finer alignment can be preformed on the classified images. Another way to deal with the misalignment problem is to modify the classification procedure in such a way that it becomes immune to the in-plane shifts. This can be achieved by the use of a shift invariant distance measure for the 1D vectors. If, for example, the distance between two 1D vectors is measured as a function of magnitudes of their Fourier transforms, then our classification method becomes inherently immune to in-plane shifts. At this point it is not clear what would be the impact of ignoring phase information on classification quality. The third approach to the misalignment problem is to incorporate a search through multiple shifted versions of the projection images into the classification method. In this approach the graph building procedure would determine the weight of each edge by finding a minimum distance between several shifted versions of the images represented by the nodes this edge connects. The computational cost associated with such search will significantly depend on the number of shifted versions of each image considered in this process. The methods for dealing with the misalignment problem are part of our ongoing research. It is possible that the best solution to this clearly nontrivial problem will be achieved by a combination of the methods proposed above.

Any attempt to reconstruct a 3D structure from its projections with unknown Euler angles in the absence of an initial model suffers from an inherent handedness problem. Two identical sets of the projection images can be produced from two different 3D objects that differ only by their spatial handedness. Since the projection angles of the images used in our classification procedure are unknown and no initial model is used, our classification procedure is unable to correctly classify heterogeneous projection set produced from two objects that differ only by handedness. (See the discussion on p. 226 of Ref. [5].)

Despite the high level of noise present in the projection images (which significantly reduces similarity between the 1D projections onto the common line), our method was able to successfully separate homogeneous subsets by the simultaneous use of the common lines between many images. Based on the histogram of edge weights (Fig. 3) it is clear that an attempt to determine whether one particular pair of images belongs to the same object using the common line approach would fail. Our method was able to

produce promising results by utilizing simultaneously all images in the set.

9. Conclusions

The results of our experiments on synthetic data sets show that, despite of the severe noise present in heterogeneous projection sets obtained by EM, partitioning (with very high classification purity) of such sets into homogeneous subsets is possible. They demonstrate that the proposed method can be successfully used in various scenarios involving heterogeneous projection sets with uneven object representation and sets containing projections of more than two objects: high quality classifications were produced at a low computational cost, illustrating that the approach is very promising for the classification of heterogeneous EM projections into homogeneous subsets.

Acknowledgment

This research is supported by the National Institutes of Health through Grant HL70472.

References

[1] J. Frank, Three-Dimensional Electron Microscopy of Macromolecular Assemblies: Visualization of Biological Molecules in their Native State, Oxford University Press, USA, 2006.
[2] P.A. Penczek, J. Frank, C.M. Spahn, J. Struct. Biol. 154 (2006) 184.
[3] J. Fu, H. Gao, J. Frank, J. Struct. Biol. 157 (2007) 226.
[4] S.H.W. Scheres, H. Gao, M. Valle, G.T. Herman, P.P.B. Eggermont, J. Frank, J.M. Carazo, Nat. Methods 4 (2007) 27.
[5] R.A. Crowther, Philos. Trans. R. Soc. Lond. Ser. B 261 (1971) 221.
[6] F. Harary, Graph Theory, Addison-Wesley, New York, 1969.
[7] V. Kann, S. Khanna, J. Lagergren, A. Panconesi, On the hardness of approximating Max k-Cut and its dual, Chicago J. Theor. Comput. Sci. (1997). ⟨http://cjtcs.cs.uchicago.edu/articles/1997/2/contents.html⟩.
[8] A. Schrijver, Combinatorial Optimization: Polyhedra and Efficiency, Springer, Berlin, 2003.
[9] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, New York, 1979.
[10] P.M. Pardalos, M.G.C. Resende, Handbook of Applied Optimization, Oxford University Press, New York, 2002.
[11] S.H.W. Scheres, M. Valle, R. Nuñez, C.O.S. Sorzano, R. Marabini, G.T. Herman, J.M. Carazo, J. Mol. Biol. 348 (2005) 139.
[12] M. Samosó, M.P. Koonce, J. Mol. Biol. 340 (2004) 1059.
[13] B.M. Carvalho, W. Chen, J. Dubowy, G.T. Herman, M. Kalinowski, H.Y. Liao, L. Rodek, L. Ruskó, S.W. Rowland, E. Vardi-Gonen, SNARK05: A Programming System for the Reconstruction of 2D Images from 1D Projections, CUNY Institute for Software Design and Development, New York, 2006. ⟨http://www.cisdd.org/snark05/SNARK05.pdf⟩.
[14] P.-N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Addison Wesley, Reading, MA, 2006.
[15] M. Kalinowski, A. Daurat, G.T. Herman, A fast construction of the distance graph used for the classification of heterogeneous electron microscopic projections, in: Proceedings of the Workshop on Graph-based Representations in Pattern Recognition, June 11–13, 2007, Lecture Notes in Computer Science 4538, Springer, Berlin, 2007, pp. 263–272.
[16] S.J. Benson, Y. Ye, X. Zhang, SIAM J. Opt. 10 (2000) 443.