A Clustering-Based Method for Particle Detection in Electron Micrographs

Zeyun Yu* and Chandrajit Bajaj†
Department of Computer Science, University of Texas at Austin
1 University Station C0500, Austin, Texas 78712-1188, USA

Abstract

Accurate and automatic particle detection (or picking) from cryo-electron microscopy (cryo-EM) images is very important for fast and correct reconstruction of macromolecular structures. In this paper, we present a new algorithm for particle picking, based on a natural model of data clustering. This approach is fully automatic and has been successfully applied to detect the particles from cryo-EM images with very low signal-to-noise ratio (SNR).

1. Introduction

Techniques from pattern recognition and image processing have been widely used in biology for molecular structure analysis. One of the examples is the macromolecular structure reconstruction from cryo-EM images. This technique for studying macromolecular structures is commonly known as *single particle reconstruction* in structural biology [1, 2]. However, the signal-to-noise ratio (SNR) in most cryo-EM images is very low due to various reasons so that high-resolution single particle analysis often has to rely on averaging of a large number of identical particles to improve the signal-to-noise ratio [1, 2]. Therefore, locating most, if not all, of the particles in the digitized cryo-EM images is a crucial step in high-resolution single particle reconstruction. This task, commonly known as particle picking or particle detection in single particle analysis, can certainly be carried out manually (by mouse clicks). However, as the resolution approaches the atomic level, hundreds of thousands of particles may be necessary [3], which makes it impractical to manually pick the particles. In addition, particle detection by eyes may be inaccurate and subjective.

Several methods have been proposed for automatic or semi-automatic particle detection (see [4] for a good review). The first automatic method for particle picking was proposed by Heel [5], based on the local variance over a small area at each pixel. Another commonly used approach is template-matching algorithm (see [6, 7, 8]), where the template is chosen as the rotationally averaged particle image and then the template is used to cross-correlate with the entire image. In [2], multiple reference templates are used to improve the accuracy of particle detection. Some other techniques, including the crosspoint method [9], the texture analysis method [10], the ring-filter based method [11], and the neural network approach [12], were proposed recently.

Another group of particle detection algorithms are based on edge detection. Harauz *et al* [13] used edge detection followed by component labeling for automatic detection of macromolecules. More recently, Zhu *et al* proposed edge detection followed by Hough transforms for automatic identification of particles with rectangular or circular shapes [14]. It is obvious that all techniques based on edge detection require a good signal-to-noise ratio. However, this is not alway true in many cryo-EM images.

In this paper we present a new method for particle picking. Our approach is based on a natural model of data clustering. In the following we first describe the clustering model and then we discuss how to apply the clustering model for particle picking. In Section 3 we shall see some results of the particle picking algorithm on real cryo-EM images. Finally we give our conclusion.

2. Approach

2.1. Data Clustering

In the following we describe a method for data clustering. Our method is based on the gravitation between any two masses as follows:

$$f(O_1; O_2) = \frac{c \times m_1 \times m_2}{d^2} \tag{1}$$

where O_1, O_2 are two objects in the space and m_1, m_2 are the masses of objects O_1, O_2, respectively. d is the Euclidean distance between these two objects in the space and c is the universal gravitation constant.

In our application, we shall assume that the input for the clustering algorithm is a 2D image consisting of background points with zero mass and a set of objects with positive masses, each of which has an XY-coordinate in the image domain. Each object generates a circular gravitation field as defined below. All such fields are integrated together, yielding an overall gravitation field. Every object moves in the direction determined by this vector field.

*email: zeyun@cs.utexas.edu
†email: bajaj@cs.utexas.edu

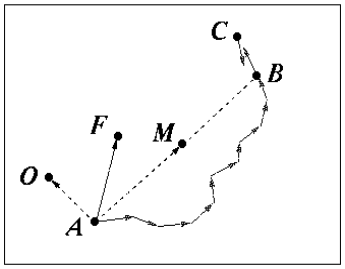


Figure 1: The illustration of the movement of an object.

Once two or more objects happen to move to the same position, they merge into a single but bigger object and will never split in the rest of the clustering process. In the following we shall discuss how to generate a gravitation field and how an object moves in a given gravitation field.

How to generate the gravitation field. Every object generates a circular gravitation field, in which the gravitation vector at any point A has the following magnitude:

$$\|\vec{f}(O;A)\| = \frac{c \times m}{d^2} \tag{2}$$

where m is the mass of the object at O and d is the distance from A to O. c is the universal gravitation constant.

From (2) we know that the gravitation decreases very rapidly to zero as d goes up. For the purpose of particle picking, we made two modifications on (2). First, we use d instead of d^2 as seen in (2) in order to enhance the "influence" of an object on other objects. Secondly, we assume the "influence zone" of an object only exists in a finite "circular zone" with a fixed radius. This assumption not only speeds up the algorithm but also avoids an embarrassing situation that all objects eventually group into one cluster.

How to determine the movement of an object. In the calculated gravitation field, each object has a path, along which it can move (e.g., the one from A to B in Fig.1). However, a very small movement of any object would change the gravitation field. A simple way for this simulation is to let every object move only one step (that is, only move to one of its neighboring pixels) and then update the overall gravitation field based on the new position of each object. Repeat these two steps alternatively until no further movement is observed. This scheme gives an accurate simulation of the movements of all the objects, but clearly it is computationally too slow due to the very small movement of each object in each iteration.

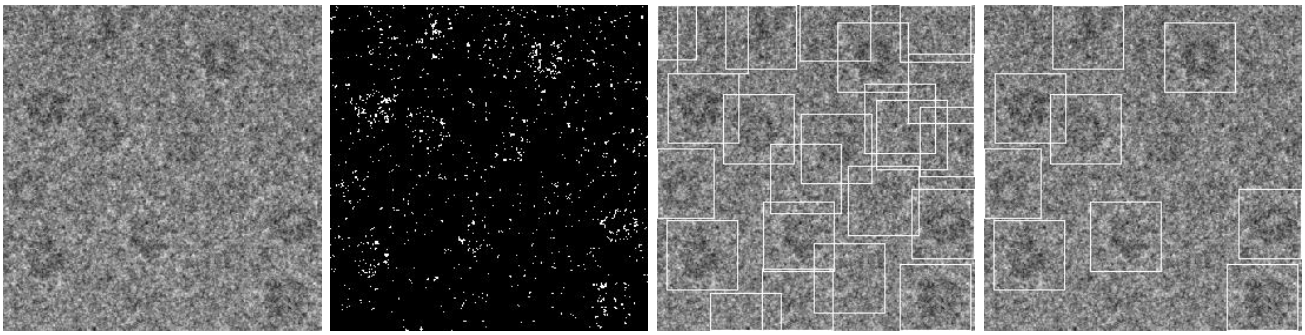
To speed up the algorithm, we can let the objects move as far as it can on the path determined by the computed gravitation field. It stops moving whenever it sees a vector that points to the opposite direction of the movement on the path. Fig.1 shows an example of this movement. The object starts from A and keeps moving until it reaches B where $\vec{v}_B \cdot \vec{v}_C \leq 0$. Then B is the farthest position, to which this object can go. The new position of this object is set to somewhere (labeled as M) on the line from A to B.

To determine the exact position of M, we need to consider three facts. First, the distance $\|\vec{AM}\|$, by which the object at A can move, should be inversely proportional to the mass of this object. This observation guarantees that the objects with small masses should always move towards the objects with much bigger masses. This is essentially important in our particle picking algorithm as seen in next section. We shall consider the relative mass. That is, $\|\vec{AM}\| \propto \frac{\bar{m}}{m_0}$, where $\bar{m}$ is the average mass of all the objects and m_0 is the mass of the object at A. $\bar{m}$ is increasing during the clustering process since the total mass is a constant but the number of objects is decreasing (due to the merging between objects). Secondly, the distance $\|\vec{AM}\|$ should be proportional to the average magnitude (denoted by $\|\vec{v}\|$) of all the vectors on the path from A to B. Thirdly, the movement $\|\vec{AM}\|$ should be no more than $0.5 * \|\vec{AB}\|$ in order to make the clustering process stable (for example, consider only two objects). Hence we have the following formula to determine the movement of an object:

$$\vec{AM} = \frac{1}{2} min(1.0, \frac{\bar{m}}{m_0} \times \|\vec{v}\|) \times \vec{AB} \tag{3}$$

Further improvements. The above algorithm can be further improved to make it much faster. First, note that in every iteration, we compute all the gravitation vectors over the entire image domain. In many applications, however, the objects are sparsely distributed and thus only very few of those vectors are used to determine the new positions of the objects. Therefore, we can compute the gravitation vector only when we need it. As soon as we compute a gravitation vector at a point, we store it so that we can use it again when we need to determine the new position of another object. This simple strategy can usually save the computational time by 90% or more. Another improvement we made is that, when we compute the new position of an object at a point A, we not only let this object move according to (3), but also let it move by a certain amount towards the "local mean" of the input data around point A. The "local mean" is defined as the weighted center of all the objects locally around A.

From the above description, we can see that our gravitation-based clustering method is quite similar to the *shift mean method* [15, 16]. However, our approach differs from the classic shift mean method in the following aspects. First, our method is a physical simulation of the motion of each data point in the gravitational force field. Second, our method allows merging between two or more data points if they run into each other at the same location in the space. This can largely reduce the number of data points being processed and thus make the clustering process very fast after the first few number of iterations.



(a) Original image (b) After threshold (c) Initial boxing (d) Final boxing

Figure 2: The illustration of particle boxing on p97 dataset [17]. In (b) we only show the binary image by assuming that all the objects have the same weights.

2.2. Particle Picking

Before we run the clustering algorithm, we need to threshold the input cryo-EM image (denoted as $f(\vec{x})$) by setting the threshold value as $h_0 = I_{min} + a * (I_{max} - I_{min})$, where I_{min} and I_{max} respectively stand for the minimal and maximal intensities of the entire image and a is set between 0 and 1. For the P97 data [17] as shown in Fig.2(a), we set $a = 0.2$. The resulting image is defined as:

$$g(\vec{x}) = \begin{cases} h_0 - f(\vec{x}) & \text{if } f(\vec{x}) \leq h_0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

In our algorithm for particle picking, we are assuming that all the particles have lower intensities than the background. Therefore, the value of $g(\vec{x})$ represents the probability of $\vec{x}$ being part of a particle. In general $g(\vec{x})$ contains more "object points" if we choose a bigger threshold value, resulting in a slower clustering process. Fig.2(b) shows the threshold result. We can see that possible particles are usually located where there are more "object points" than elsewhere. The clustering approach described above is then applied to the image $g(\vec{x})$ and the obtained centers give an initial particle picking result, as seen in Fig.2(c).

To reduce the false positives, we usually need to evaluate each of the initially-detected particles and remove the "false" ones. We currently use two criteria to determine whether a particle is true or not. The first criterion is the local intensity variance σ^2 of a particle within its box (we assume the size of box is given and fixed). If the local intensity variance is smaller than a given value, this particle is removed from the particle list. The second criterion is based on the requirement that the true particles must be located around the center of the boxes. If a particle is too far away from the center of its box, this particle is also recognized as false particle. To measure how far a particle is away from the center of the box, we compute the average intensity within a circle around the center where the size of the circle is chosen exactly the same as the size of the particles. We also compute the average intensity of all the other pixels that are outside of the circle but still inside the box. If these two averages are too close to each other, this particle is also removed from the particle list. Fig.2(d) shows the results after refinement.

3. Results

Due to the space limit, we show only one example of our particle picking algorithm on P97 datasets [17] (seen in Fig.3). The size of each box is 80 pixels and the number of iterations used in the clustering process is 30. More results can be found from the following website: "http://www.ices.utexas.edu/∼zeyun/PtcPick/".

4. Conclusion

In this paper we described a new model for data clustering, which was then used to detect particles from electron micrographs. The results on P97 data showed that our method worked quite well for detecting particles from images with very low SNR.

References

[1] J.Frank, *Three-dimensional Electron Microscope of Macromolecular Assemblies*, San Diego, Academic Press, 1996.

[2] S.J. Ludtke, P.R. Baldwin and W.Chiu, "EMAN: semiautomated software for high-resolution single-particle reconstructions", *J. of Structural Biology*, vol. 128, pp. 82-97, 1999.

[3] R. Henderson, "The potential and limitations of neutrons, electrons and X-rays for atomic resolution microscopy of unstained biological macromolecules", *Quart. Rev. Biophys*. vol. 28, pp. 171-193, 1995.

[4] W.V. Nicholson and R.M. Glaeser, "Review: automatic particle detection in electron microscopy", *Journal of Structural Biology*, vol. 133, pp. 90-101, 2001.

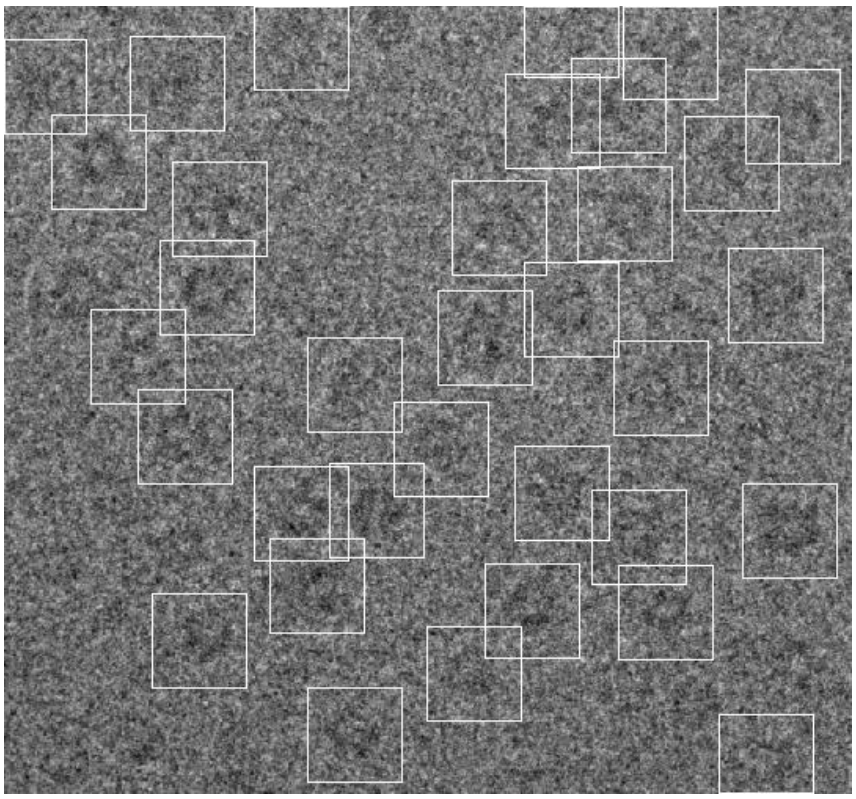


Figure 3: One example of particle picking on P97 dataset. Box size: 80 pixels. Number of iterations for clustering: 30

[5] M. van Heel, "Detection of objects in quantum-noise-limited images", *Ultramicroscopy*, vol. 7, pp. 331-342, 1982.

[6] J. Frank and T. Wagenknecht, "Automatic selection of molecular images from electron micrographs". *Ultramicroscopy*, vol. 12, pp. 169-176, 1984.

[7] A. Saad, W. Chiu and P. Thuman-Commike, "Multiresolution approach to automatic detection of spherical particles from electron cryomicroscopy images". *Proc. of IEEE Intl. Conf. on Image Proc.*, Chicago, IL, pp. 8-10, 1998.

[8] P. Thuman-Commike and W. Chiu, "PTOOL: a software package for the selection of particles from electron cryomicroscopy spot-scan images". *Journal of Structural Biology*. vol. 116, pp. 41-47. 1996.

[9] I.M.B. Martin, D.C. Marinescu, R.E. Lynch and T.S. Baker, "Identification of spherical virus particles in digitized images of entire electron micrographs". *Journal of Structural Biology*, vol. 120, pp. 146-157, 1997.

[10] K.R. Lata, P. Renczek and J. Frank, "Automatic particle picking from electronic micrographs", *Ultramicroscopy*, vol. 58, pp. 381-391, 1995.

[11] T. Kivioja, J. Ravantti, A. Verkhovsky, E. Ukkonen and D. Bamford, "Local average intensity-based method for identifying spherical particles in electron micrographs", *J. of Structural Biology*, vol.131, pp.126-134, 2000.

[12] T. Ogura and C. Sato, "An automatic particle pickup method using a neural network applicable to low-contrast electron micrographs", *Journal of Structure Biology*, vol. 136, pp. 227-238, 2001.

[13] G. Harauz and A. Fonf-Lochovsky, "Automatic selection of macromolecules from electron micrographs by component labeling and symbolic processing", *Ultramicroscopy*, vol. 31, no. 4, pp. 333-344, 1989.

[14] Y. Zhu, B. Carragher, D. Kriegman, R. Milligan and C. Potter, "Automated identification of filaments in cryo-electron microscopy images", *J. Str. Bio.*, vol. 135, pp.302-312, 2001.

[15] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 1-18, 2002.

[16] Y. Cheng, "Mean shift, model seeking and clustering", *IEEE Trans. on PAMI*, vol. 17, no. 8, pp. 790-799, 1995.

[17] I. Rouiller, J. Pulokas, V. Butel, R. Milligan, E. Wilson-Kubalek, C. Potter and B. Carragher, "Automated image acquisition for single-particle reconstruction using p97 as the biological sample", *J. Str. Bio.*, vol. 133, pp.102-107, 2001.