中國科學院

计算数学与科学工程计算研究所

# A Fast Classification Method for Single-Particle Projections with a Translation and Rotation Invariant

by

**Xia Wang and Guoliang Xu**

Report No. **ICM-12-01** March 2012

## Research Report

**Institute of Computational Mathematics**
**and Scientific/Engineering Computing**
**Chinese Academy of Sciences**

# A Fast Classification Method for Single-Particle Projections with a Translation and Rotation Invariant[*]

Xia Wang Guoliang Xu [†]

LSEC, Institute of Computational Mathematics and Scientific/Engineering,Academy of
Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China

March 9, 2012

**Abstract**

The aim of the electron microscopy image classification is to categorize the projection images into different classes according to their similarities. Distinguishing images usually requires that these images are aligned first. However, alignment of images is a difficult task for a highly noisy data set. In this paper, we propose a translation and rotation invariant based on the Fourier transform for avoiding alignment. A novel classification method is therefore established. To accelerate the classification speed, secondary-classes are introduced in the classification process. The test results also show that our method is very efficient and effective. Classification results using our invariant are also compared with the results using other existing invariants, showing that our invariant leads to much better results.

**Key Words**: Classification; Fourier transform; Translation and rotation invariant; Secondary-class.

## 1 Introduction

Single-particle reconstruction (SPR) is a powerful method in three dimensional electron microscopy [2], which demands that all of the projection images are from the nearly identical macromolecular “particles”. The aim of the SPR is to find three dimensional structures

[*] Project support in part by NSFC under the grant 60773165, NSFC key project under the grant 10990013 and Funds for Creative Research Groups of China (grant No. 11021101).

[†] Corresponding author.

*E-mail address*: xuguo@lsec.cc.ac.cn

of a macromolecule given its two-dimensional noisy projection images at unknown random directions [15].

One of the main problems in electron microscopy is that the object is damaged by the exposure. To avoid the damage, low electron dose is used, and therefore the projection images exhibit very low signal to noise ratios (SNR, below 1/3) and very poor contrast [16]. A way to solve this problem is to put many identical objects onto the stage, which results in many projection images of the same object at unknown orientations [10]. In order to reduce the effect of the high noise and poor contrast, a large number of projection images (from $10^4$ to $10^6$) are collected and analyzed, which means that the computation load is heavy even for a modern multi-processor computer cluster [17]. Therefore, a primary step of the single-particle analysis is the classification of the measured images according to their similarities. Images in the same class are then averaged to reduce the noise level [3, 7].

**Previous Work**

Classification is a procedure that categorizes images into different classes according to their similarities [9]. Existing classification methods are divided into supervised classification and unsupervised classification [2]. Supervised classification is to categorize the images according to the similarity with templates or references, for example, projection matching. Unsupervised classification is to classify the images according to their intrinsic differences. According to the comparison among 2-D projection images, these projection images are classified into different classes. Distinguishing among different image classes requires that the images in these classes are aligned first. As a result, the computational load is heavy because any two projection images need to be aligned before their correlations are computed. Alignment is a difficult task since it is hardly possible to align a highly noisy data set without any deviation [13].

Van Heel et al. [8] propose a multireference alignment method in which the alignment and classification steps are iteratively alternated until convergence. Assuming that a set of class representatives have been selected, the similarities between the aligned images and the representatives are measured. The image is assigned to the class with the maximum similarity. Eventually, the representative of the class is recomputed as the average of the images assigned to the class. This process is repeated till some convergent criterion is met. A possible shortcoming of this classification method is that it depends on the initial selection of the class representatives and it possibly traps in a local minimum. To solve these problems, a multireference alignment algorithm based on a maximum likelihood (ML2D) was devised in [8]. However, ML2D [14] method suffers from the *attraction problem* [16], a phenomena that a class-average with less noise than others attracts more experimental images even if they belong to other classes.

Colars et al. [16] propose a novel method of clustering 2D (CL2D) images that is able to address small differences between classes. With this method, all the projection images can be split into specified number classes and at the same time the misclassification error is minimized. The key ideas of the CL2D method are as follows: Firstly, the correlation is replaced by the correntropy, a more effective metric. Secondly, the images are assigned to the class by considering the images are more suitable for the class representative than the other experimental images. This comparison avoids comparing an experimental image to the class averages at the different noise levels.

Our classification method is basing on a translation and rotation invariant. We therefore review here a few existing translation and rotation invariants, which have been used in electron microscopy [13, 12]. The detail descriptions of these invariants are presented in the appendix of this paper. Van Heel et al. introduce in [12] a double auto-correlation function (briefed as DACF) to classify the projection images. At first, the auto-correlation functions (ACF) of the projection images are calculated and converted to the cylindrical coordinates. Then the second ACFs in the angular direction of the ACFs in cylindrical coordinates are computed, resulting in the DACFs. The authors in [12] point out that DACFs is overweight the already strong frequency because of the squaring of the Fourier components in calculating the ACF. This disadvantage can be eliminated by using self-correlation function (SCF) instead of auto-correlation function. In the literatures [1, 4], the authors present a complete rotation invariant (an invariant without losing phase information), named as AFMT. AFMT is defined using the analytical Fourier-Mellin transform and angular corrections for the centralized images.

**Our Contributions**

In this paper we propose an classification method based on the application of a translation and rotation invariant of two-dimensional images to avoid performing image alignment. The main ideas of our classification method are as follows: (a) We propose a translation and rotation invariant based on the combing use of the Fourier transform. (b) We classify the projection images with an adaptive threshold $\varepsilon$, such that our class-sizes (diameter) in space are as close to uniform as possible, avoiding the *attraction problem* even at low SNR. (c) We introduce a secondary-class along with each class, which greatly accelerates the process of class adjusting. (d) We propose a proper way to cut off the high frequency in the Fourier space making our method anti-noise and robust.

The organization of this paper is as follows. In Section 2, we give the related mathematical background of our method. In Section 3, we introduce the used notations and definitions, followed with the classification method of the projection images, together with

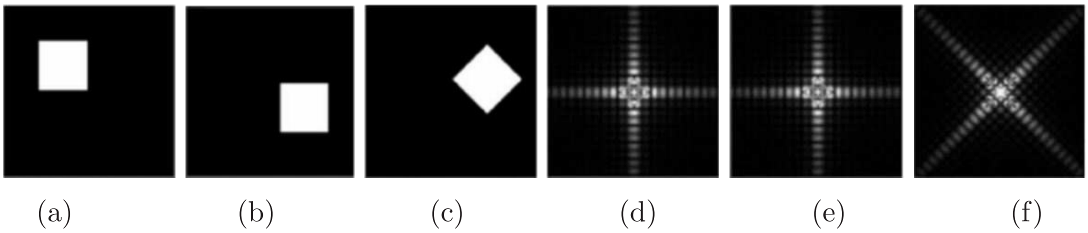


Fig 1: Figure (a) an image of a square. Figure (b) a translated image of (a). Figure (c) a translated and rotated version of (a). Figures (d), (e) and (f) show the Fourier spectra of (a), (b) and (c), respectively.

the implementation details. Experimental results for two simulated data sets are given in Section 4. We conclude the paper with a summary in Section 5. The detail descriptions of several other translation and rotation invariants are presented in Appendix A.

## 2 Preliminary Mathematical Material

In this section, we introduce the used preliminary mathematical material, including the continuous translation and rotation invariant, the truncated translation and rotation invariant, the spectrum power at the low frequency and the analytical Fourier-Mellin transform.

**Continuous Translation and Rotation Fourier Invariant**. Our classification method relies on the translation and rotation property of the Fourier transform [11]. Let $f_1$ be an image and $f_2$ the translated image of $f_1$ with translation $(x_0, y_0)$, i.e.,

$$f_2(x, y) = f_1(x - x_0, y - y_0). \tag{1}$$

Then it is well known that the corresponding Fourier transforms $F_1$ and $F_2$ have the following relationship

$$F_2(\xi, \eta) = e^{-i2\pi(\xi x_0 + \eta y_0)} F_1(\xi, \eta), \tag{2}$$

where $i$ is the imaginary unit. Hence, the Fourier spectra of $f_1$ and $f_2$ are the same. Fig 1 shows the translation property of the Fourier transform (in order to observe the similarities and differences of the Fourier spectra, the Fourier spectra in all the figures, except for Fig 2, are normalized into the range $[0, 2550]$ and then mod 255. In Fig 2, Fourier spectrum is normalized into the range $[0, 255]$).

When an image is rotated, the image of the Fourier spectrum is also rotated with the same angle. If $f_2(x, y)$ is a translated and rotated function of $f_1(x, y)$ with a translation

$(x_0, y_0)$ and a rotation angle $\theta_0$, then we have

$$f_2(x,y) = f_1(x\cos\theta_0 + y\sin\theta_0 - x_0, -x\sin\theta_0 + y\cos\theta_0 - y_0).$$

Fig 1 shows the Fourier rotation property. According to the Fourier translation and rotation property, the Fourier transforms of $f_1$ and $f_2$ are related by

$$F_2(\xi,\eta) = e^{-i2\pi(\xi x_0+\eta y_0)} F_1(\xi\cos\theta_0 + \eta\sin\theta_0, -\xi\sin\theta_0 + \eta\cos\theta_0). \tag{3}$$

Let $M_1$ and $M_2$ be the magnitude functions of $F_1$ and $F_2$. Then from equation (3), we have

$$M_2(\xi,\eta) = M_1(\xi\cos\theta_0 + \eta\sin\theta_0, -\xi\sin\theta_0 + \eta\cos\theta_0).$$

It is easy to see that the magnitude functions of both the spectra are the same, but one is a rotated version of the other. Rotational movement without translation can be represented as a translational displacement in polar coordinates. i.e., in the polar representation

$$M_2(\rho\cos\theta, \rho\sin\theta) = M_1(\rho\cos(\theta-\theta_0), \rho\sin(\theta-\theta_0)).$$

It is denoted as

$$\widetilde{M_2}(\rho,\theta) = \widetilde{M_1}(\rho,\theta-\theta_0). \tag{4}$$

Following equation (1) and (2), the modules of the corresponding Fourier transforms of $\widetilde{M_1}$ and $\widetilde{M_2}$ are the same. Therefore, the module is a translation and rotation invariant. Replacing the polar coordinate by the log-polar coordinate, we obtain a similar translation and rotation invariant.

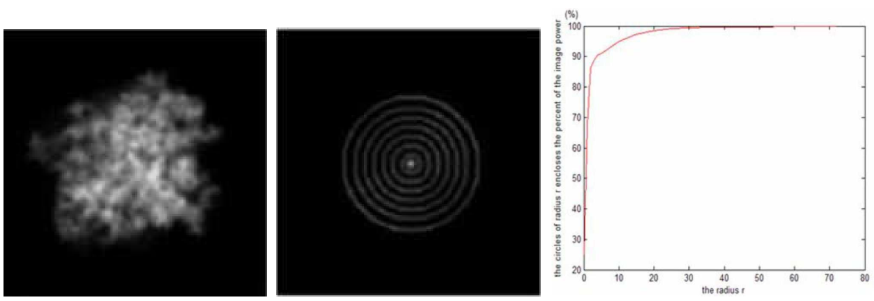
**The Spectrum Power at the Low Frequency**. We present a method to determine the cut-off frequency by computing the circles that enclose specified amounts of total image power $P_T$ (see [6]). The total image power for an $N \times N$ image $I(u,v)$ is defined as

$$P_T = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} P(u,v),$$

where $P(u,v) = [\mathrm{real}(I(u,v))]^2 + [\mathrm{imag}(I(u,v))]^2$ is the spectrum power of the image $I(u,v)$.

Fig 2 shows a projection image with size 143 × 143 and the Fourier spectrum of the image. The Fourier transform has been centralized and a circle of radius $r$ with the origin at the center of the frequency domain encloses $\alpha$ percent of the power, where

$$\alpha = \sum_{u^2+v^2\le r^2} P(u,v)/P_T.$$



(a) (b) (c)

Fig 2: (a) is a projection image of size 143 × 143 pixels and (b) is its Fourier spectrum. The superimposed circles of radii $r = 5, 10, 15, 20, 25, 30$ and 36 enclose 91.07%, 94.97%, 97.42%, 98.46%, 99.07%, 99.43% and 99.68% of the image power, respectively. (c) shows the percentage of the image power enclosed by the circle of radii $r$.

In (b) of Fig 2, we take the radii of $r = 5, 10, 15, 20, 25, 30$ and 36 pixels, respectively. Each of the circles encloses $\alpha$ percent of the image power with $\alpha = 91.07\%$, 94.97%, 97.42%, 98.46%, 99.07%, 99.43% and 99.68%, respectively. Therefore, if the frequency components are normalized into $[-\pi, \pi]$, we take the cut-off frequency as $-\pi/2$ and $\pi/2$ and more than 99.6% of the total power is enclosed. From this point of view, the cut-off frequency can be required to belong to $[-\pi/2, \pi/2]$.

If the continuous Fourier transform is replaced by the discrete Fourier transform, equation (4) is still true at the frequencies in $[-\pi/2, \pi/2]$ and enough information is kept at these frequencies. Hence, the spectrum from double FFT is also a translation and rotation invariant in the range $[-\pi/2, \pi/2]$.

**The Analytical Fourier-Mellin Transform**. Let $f(t, \theta)$ be the Fourier spectrum in the polar coordinate $(t, \theta)$ of image $I$. Then the analytical Fourier-Mellin transform (AFMT) of $f$ is defined as

$$F(\omega, \nu) = \int_0^R \int_0^{2\pi} f(t, \theta) t^{\sigma - i\omega} e^{-i\theta\nu} d\theta \frac{dt}{t},$$

where $\sigma$ is a fixed and positive real number. In our algorithm, we weight the Fourier spectrum in polar (or log-polar) coordinate with $r^\sigma$. This means that the analytical Fourier-Mellin transform is used. Following the suggestion of Goh [5], $\sigma$ is set to 0.5.

## 3 Classification Algorithms

In this section, we first introduce some notations and definitions, then we present the classification algorithm outlines and the implementation details.

## 3.1 Notations and Definitions

**Fourier Transform Based Translation and Rotation Invariant**. Firstly, the Fourier spectra of the given images are computed. Then, the conversion from Cartesian into the polar (or log-polar) coordinate is performed and then these spectra in the polar (or log-polar) coordinate are weighted. Finally, the Fourier spectra of these weighted spectra in the polar (or log-polar) coordinate are computed and the latter spectra are weighted again. The final results are called the Fourier transform based translation and rotation invariants (briefed as FTTR-invariants). Throughout this paper, we denote by $I_i$ the $i$-th projection image and $\widetilde{I}_i$ the FTTR-invariant of $I_i$.

**Similarity Metric of Two Images**. Let $I_i$ and $I_j$ be any two $N \times N$ matrices representing the experimental images. Let $\tilde{I}_i$ and $\tilde{I}_j$ of size $P \times P$ ($P \leq N$) be the FTTR-invariants of $I_i$ and $I_j$, respectively. We denote by $d(I_i, I_j)$ the similarity metric (distance) of two images $I_i$ and $I_j$, defined as

$$d(I_i, I_j) := d(\tilde{I}_i, \tilde{I}_j) = \|\tilde{I}_i - \tilde{I}_j\|_F,$$

where $\|\cdot\|_F$ stands for the Frobenius norm of a matrix.

**Classes**. For a given $\varepsilon$, the class $\mathbb{C}_i$ and its representative $R_i$ are defined as follows: $\mathbb{C}_i$ is a collection of images and $R_i$ is a member of $\mathbb{C}_i$, such that

$$d(I_k, R_i) < \varepsilon, \quad \forall I_k \in \mathbb{C}_i.$$

Let $\widetilde{\mathbb{C}}_i$ be a set of the invariants of all the projection images in $\mathbb{C}_i$, named as invariant-class. We use $|\widetilde{\mathbb{C}}_i|$ to denote the cardinality of the invariant-class $\widetilde{\mathbb{C}}_i$. The invariant-class center $C_i$ is defined as

$$C_i = \frac{1}{|\widetilde{\mathbb{C}}_i|} \sum_{\widetilde{I}_j \in \widetilde{\mathbb{C}}_i} \widetilde{I}_j. \tag{5}$$

**Secondary-Class**. Suppose a set of 2D images $\{I_i\}_{i=1}^n$ of size $N \times N$ has been assigned into $M$ classes. Let

$$\gamma = \max_{i=1,\cdots,M} \max_{I_k \in \mathbb{C}_i} d(\widetilde{I}_k, C_i), \tag{6}$$

where $C_1, C_2, \cdots, C_M$ are the invariant-class centers defined by (5). Then $d(\widetilde{I}_k, C_i) \leq \gamma$ for $I_k \in \mathbb{C}_i,\ i = 1, \cdots, M$. The secondary-class $\mathbb{S}_i$ associated with the class $\mathbb{C}_i$ consists of all the invariant-class centers $C_j$, such that $d(C_j, C_i) \leq 2\gamma$. Note that each $C_i$ may belong to several secondary-classes.

## 3.2 Algorithms Outlines

Suppose we are asked to classify $\{I_i\}_{i=1}^n$ into $M$ classes. In the following, we present algorithm outlines for achieving this goal. The detail descriptions are given in the next subsection.

**Algorithm 3.2.1**. *Compute the FTTR-invariants*

For $i = 1, 2, \cdots, n$, do the following:

1. Apply discrete Fourier transform to all the projection images $I_i$, obtaining $F_i$. $F_i$ is computed using FFT. Sometimes we need to expand the sizes of all the projection images by adding zeros before performing the FFT to raise the classification accuracy.

2. Compute the module $M_i$ of $F_i$, and then find the valid area of $M_i$ (see Section 3.3). Assume the size of the valid area is $N_1 \times N_1$.

3. Convert $M_i$ from Cartesian to the polar (or log-polar) coordinate form in the valid area. The polar (or log-polar) value is weighted by $r^{\sigma_1}$ ($\sigma_1 > 0$). Then compute the discrete Fourier transform by FFT of the weighted polar form, obtaining an image $W_i$.

4. Set the valid area of $W_i$ as $[-N_1/2, N_1/2] \times [-N_1/2, N_1/2]$. Compute the module of $W_i$ in their valid area. The modules are weighted by $(\frac{\sqrt{2}N_1}{2} - r)^{\sigma_2}$ ($\sigma_2 > 0$). The output of this step is our FTTR-invariant $\tilde{I}_i$.

The first weight $r^{\sigma_1}$ in step 3 plays a key role to the success of the classification algorithm. Following the suggestion of Goh [5], $\sigma_1$ is set to 0.5 in our implementation. The second weight $(\frac{\sqrt{2}N_1}{2} - r)^{\sigma_2}$ is less important to the success of the classification and it makes the classification result a little better. We require $0 < \sigma_2 \leq 2$. Fig 3 shows our invariants of the projection images only with translation and rotation difference and Fig 4 shows the invariants of the different projection images. Since a valid area for the invariants in the Fourier space is used in our algorithm, the invariants are antinoise.

**Algorithm 3.2.2**. *Classification with the FTTR-invariants.*

1. Compute an initial value of $\varepsilon$. Set the iteration number $s = 0$.

2. Classify the projection images controlling with the $\varepsilon$. Suppose we get $\nu$ classes $\mathbb{C}_1, \mathbb{C}_2, \cdots, \mathbb{C}_\nu$. Compute $\delta$ by

$$\delta = \max_{i=1,\cdots,\nu} \max_{I_k \in \mathbb{C}_i} d(I_k, R_i). \tag{7}$$

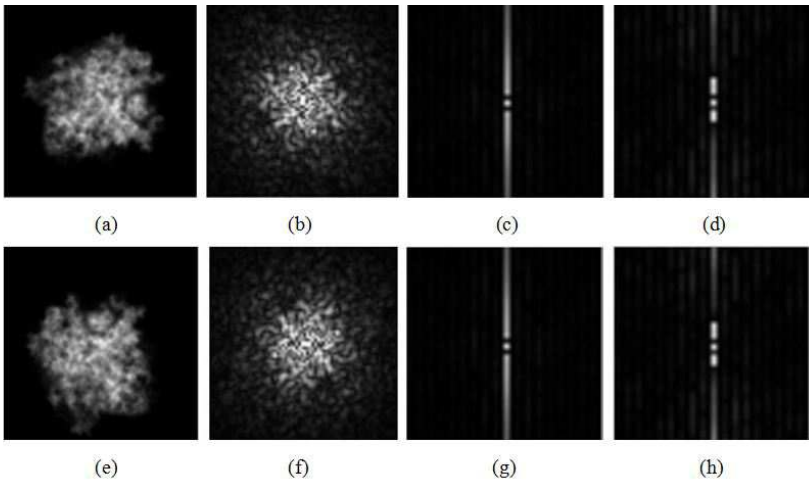


Fig 3: (a) is a projection image. (e) is the rotated version of (a). (b) and (f) are Fourier spectra of (a) and (e), respectively. (c) and (g) are Fourier spectra of (b) and (f) in polar coordinate, respectively. (c) and (g) are weighted and become (d) and (h). (d) and (h) are the FTTR-invariants for classification.

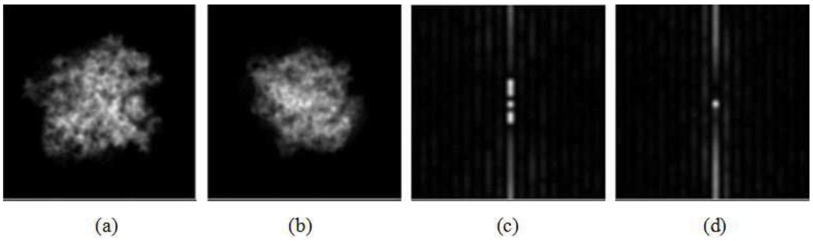


Fig 4: (a) and (b) are different projection images. (c) and (d) are respectively the invariants of (a) and (b).

3. If $\nu < aM$ $(0 < a < 1)$ or $\nu > bM$ $(b > 1)$, and $s \leq S$ (we take $S$ as 10), compute a new $\varepsilon$ using the computed $\nu$ and $\delta$. Set $s$ as $s + 1$. Then go back to step 2.

4. If $aM \leq \nu < M$, split the large classes until $\nu = M$. If $M < \nu \leq bM$, merge the two classes with the smallest distance until $\nu = M$.

5. Adjust the classification using the minimum distance between the projection images and the invariant-class center in the secondary-classes of its assigned class.

6. Find class representative or class average (optional).

In the third step, $a$ $(a < 1)$ and $b$ $(b > 1)$ are real numbers close to 1. The closer $a$ and $b$ are to 1, the more uniform the classification result is. However, it causes the more computation load for computing a new $\varepsilon$. In our experiments, we choose $a = 0.95, b = 1.05$.

In the sixth step, the representative of a class is the projection image such that its FTTR-invariant is nearest to the invariant-class center in equation (5). The class average can be computed using the public software xmipp_average by aligning the images in the same class in the real space. Fig 9, Fig 14 and Fig 17 show several class averages of our classification results.

### 3.3 Implementation Details

In this subsection, we provide the implementation details of our algorithms presented in the previous subsection.

**Compute an initial** $\varepsilon$. Randomly choose $p$ projection images and compute their distance between each other. The average of these distances is taken as the initial value of $\varepsilon$. We wish to have an initial value of $\varepsilon$ that can better represent the general distance between the projection images. It is not necessary to choose a large $p$. However, if $p$ is too small, the initial value of $\varepsilon$ is not representative. In our experiments, we choose $p = M$.

**Compute the valid area**. The valid area in frequency domain in Algorithm 3.2.1 is computed as follows.

1. Compute the average of $\{M_i\}_{i=1}^n$ in step 2 of Algorithm 3.2.1. $\bar{M} = \frac{1}{n}\sum\limits_{i=1}^{n} M_i$.

2. From $R = 1$ to $R = N/2$, compute the average of $\bar{M}$ out of the circle of radius $R$ and denote the average as $\mu_R$, where $N \times N$ is the size of $\bar{M}$.

3. From $r = 1$ to $r = R$, compare the magnitude of $\mu_R$ and the module on the circle of radius $r$ till some module less than $\mu_R$ appears. Denote this $r$ as $D_R$. Let $D_0 = \frac{2}{N}\sum\limits_{R=1}^{N/2} D_R$. If the frequency components are normalized into $[-\pi, \pi]$, our cut-off frequency is taken as $-D$ and $D$, where $D = min\{\frac{2\pi D_0}{N}, \frac{\pi}{2}\}$. The reason that the cut-off frequency belong to $[-\pi/2, \pi/2]$ has been explained in Section 2. Let $D_0 = \frac{DN}{2\pi}$. Then $[-D_0, D_0] \times [-D_0, D_0]$ is taken as the first valid area in the second step of Algorithm 3.2.1.

**Classification**. Controlling with an $\varepsilon$, we classify the projection images progressively. At beigining, $I_1$ is put into the first class. Suppose we have classified $I_1, I_2, \cdots, I_{k-1}$ into classes $\mathbb{C}_1, \mathbb{C}_2, \cdots, \mathbb{C}_m$. Suppose $R_i$ is the representative of $\mathbb{C}_i$, $i = 1, 2, \cdots, m$. Next we determine which class $I_k$ belongs to. Compute the distance between $\widetilde{I}_k$ and $\widetilde{R}_i$, for $i = 1, 2, \cdots, m$.

1. If $\min_{i=1,\cdots,m} d(\widetilde{I}_k, \widetilde{R}_i) < \varepsilon$, then $I_k$ is assigned into $\mathbb{C}_\beta$ with $\beta = \arg\min_{i=1,\cdots,m} d(\widetilde{I}_k, \widetilde{R}_i)$.

2. If $\min_{i=1,\cdots,m} d(\widetilde{I}_k, \widetilde{R}_i) \geq \varepsilon$, then $I_k$ is defined as the representative of the $(m+1)$-th class, a newly created class.

Repeating the procedure above for $k = 2, \ldots, n$, we obtain a classification $\{\mathbb{C}_j\}_{j=1}^{\nu}$ of all the images and a $\delta$ computed by equation (7).

**Compute a new** $\varepsilon$. According to the classification described above, given an $\varepsilon$, we get $\nu$ classes. Obviously, the bigger $\varepsilon$ leads to smaller $\nu$. If $\varepsilon$ is too large or too small, $\nu$ is far away from the desired class-number $M$. Therefore, a new $\varepsilon$ needs to be computed according to the previously computed $\nu$ and $\delta$. We compute the new $\varepsilon$ by the bisection method from previous $\delta$ and $\nu$.

**Split classes**. If $aM < \nu < M$, the largest classes are split. Let $|\mathbb{C}_i|$ be the cardinality of class $\mathbb{C}_i$. Sorting $|\mathbb{C}_1|, |\mathbb{C}_2|, \cdots, |\mathbb{C}_\nu|$ from big to small, we get

$$|\mathbb{C}_{s_1}| \geq |\mathbb{C}_{s_2}| \geq, \cdots, \geq |\mathbb{C}_{s_\nu}|.$$

Then we split $\mathbb{C}_{s_i}, i = 1, 2, \cdots, M - \nu$ as follows:

1. Find a projection $I_i \in \mathbb{C}_{s_i}$ such that its FTTR-invariant $\tilde{I}_i$ is farthest away from $C_{s_i}$ and then find $I_{\nu+i} \in \mathbb{C}_{s_i}$ that is farthest away from $I_i$ in $\mathbb{C}_{s_i}$.

2. Taking $I_i$ and $I_{\nu+i}$ as the representatives of two newly created classes, all the projection images in $\mathbb{C}_{s_i}$ are reassigned into the new classes $\mathbb{C}_{s_i}$ and $\mathbb{C}_{\nu+i}$ with the minimal distance.

By this splitting process, we split $\mathbb{C}_{s_i},\ i = 1, 2, \cdots, M - \nu$, and we obtain classes $\mathbb{C}_i,\ i = 1, 2, \cdots, M$. After the initial classification, the class-size (diameter) is within $\varepsilon$. Hence, we use the number of the class elements instead of the class-size as a split condition.

**Merge classes**. If $M < \nu < bM$, the closest classes are merged. This is done as follows.

1. Compute
$$d(C_i, C_l),\ i = 1, 2, \cdots, \nu - 1;\ l = i + 1, i + 2, \cdots, \nu.$$

2. Sorting $\{d(C_i, C_l)\}$ in the increasing order, we get
$$d(C_{i_1}, C_{l_1}) \leq d(C_{i_2}, C_{l_2}) \leq, \cdots, \leq d(C_{i_L}, C_{l_L}),\ L = \nu(\nu - 1)/2.$$

3. For $k = 2, \cdots, L$, delete the elements of $\{d(C_{i_k}, C_{l_k})\}_{k=2}^{L}$ repeatedly as follows: If $d(C_{i_k}, C_{l_k}) \in \{d(C_{i_k}, C_{l_k})\}_{k=1}^{L}$ is an element undeleted, then we delete all the element $d(C_{i_j}, C_{l_j})$ if
$$j > k \quad \text{and} \quad \{i_k, l_k\} \cap \{i_j, l_j\} \neq \emptyset.$$
After the deleting, we obtain a new increasing sequence $\{d(C_{p_k}, C_{q_k})\}_{k=1}^{L_1}$ such that $\{p_i, q_i\} \cap \{p_j, q_j\} = \emptyset$ $(i \neq j)$.

4. Reassign all the projection images in the class $\mathbb{C}_{q_k}$ into the class $\mathbb{C}_{p_k}$, for $k = 1, 2, \cdots, \nu - M$. Then delete the class $\mathbb{C}_{q_k}$, $k = 1, 2, \cdots, \nu - M$. Finally, we obtain $\mathbb{C}_i, i = 1, 2, \cdots, M$.

**Adjust classes**. If $\nu = M$, we begin to adjust the classification with the help of the secondary-class defined in Section 3.1. Since our secondary-class $\mathbb{S}_i$ records the invariant-class centers near the $i$-th class $\mathbb{C}_i$, adjusting the classification is performed in $\mathbb{S}_i$. This is done as follows:

1. Compute the secondary-class $\mathbb{S}_i$ for $i = 1, \cdots, M$.

2. For each $I_j \in \mathbb{C}_i$, do the following:

   (a) Compute $q$ such that $C_q = \arg \min_{C_k \in \mathbb{S}_i} (d(\widetilde{I}_j, C_k))$.

   (b) If $d(\widetilde{I}_j, C_q) < d(\widetilde{I}_j, C_i)$, reassign $I_j$ to the class $\mathbb{C}_q$.

3. If one of the following two terminating conditions is satisfied, stop the iteration. Otherwise, go back to step 1.

   (a) the iteration-number reaches a threshold (we take it as 20).

   (b) the ratio of the number of the adjusted images to the total number of all projection images is lower than a fixed percentage (we take it as 1%).

**Compute the secondary-classes**. The secondary-classes are computed as follows.

1. Compute the invariant-class centers $C_1, C_2, \cdots, C_M$ by (5).

2. Compute $\gamma$ by equation (6).

3. Compute the distances $d(C_i, C_j)$, $i = 1, 2, \cdots, M-1$; $j = i+1, i+2, \cdots, M$. If $d(C_i, C_j) \leq 2\gamma$, $C_j$ is put into $\mathbb{S}_i$ and $C_i$ is put into $\mathbb{S}_j$. We then obtain the secondary-classes $\mathbb{S}_1, \mathbb{S}_2, \cdots, \mathbb{S}_M$.

**Remark 3.1.** *After the class adjustment, there may appear empty classes. Hence it is possible that the final number of classes may be less than the specified number $M$. However, since the initial classification is almost uniform in class-size, the adjustment is performed locally. Therefore, the final class number is close to the required number.*

**Remark 3.2.** *The reason why we choose 2$\gamma$ as the radius of the secondary-class is that if $d(C_i, C_j) > 2\gamma$ and $I_\alpha \in \mathbb{C}_i$, then it is impossible that $I_\alpha$ is reassigned into $\mathbb{C}_j$ in the classification-adjusting. This fact is proved as follows. Using the triangle inequality, we have*

$$d(C_i, C_j) \leq d(C_i, \widetilde{I}_\alpha) + d(\widetilde{I}_\alpha, C_j).$$

*Since $I_\alpha \in \mathbb{C}_i$ and $\widetilde{I}_\alpha$ are the FTTR-invariants of $I_\alpha$, we have $d(C_i, \widetilde{I}_\alpha) \leq \gamma$. Therefore,*

$$d(\widetilde{I}_\alpha, C_j) \geq d(C_i, C_j) - d(\widetilde{I}_\alpha, C_i) > \gamma \geq d(\widetilde{I}_\alpha, C_i).$$

*Thus, it is impossible that $I_\alpha$ can be reassigned into $\mathbb{C}_j$ in the classification-adjusting.*

## 4 Experimental Results

To validate our algorithm, we use the same simulated datasets as Colars et.al [16]. One is the bacteriorhodopsin monomer. The other is the Escherichid coli ribosome. In our experiments, we compare our results to those of CL2D [16]. We obtain similar or better angular results with much less computational time. In CL2D classification, we use the correntropy and the robust classification criterion. In all the experiments, the classification is adjusted with a maximum iteration count of 20 or with adjusted image-number less than 1% of the total number of the projection images. All the experiments are conducted on a computer cluster in our institute with a specified number of nodes. There are 8 cores per node (Intel(R) Xeon(R), 2.40GHz).

CL2D has been shown to be an efficient method. It yields better experiment results than the other well known methods, such as ML2D, PCA/K-means and PCA/Hierarchical, especially for the data set with low SNR. Therefore, we compare in this paper the performance of our method with that of CL2D.

### 4.1 Simulated Data: Bacteriorhodopsin with SNR=0.83

To illustrate our algorithm is antinoise, we add white Gaussian noise to the projection images of Bacteriorhodopsin of size 143 × 143 with SNR=0.83 (see Fig 5). Since the SNR is not very low, we do not need any filter and apply our algorithm directly to classify 5000 projection images into 256 classes. We obtain a similar result to the classification of the images without noise. Fig 6 shows the images of one class from our classification result. Fig 7 shows all the images in Fig 6, but without adding noise. We can see that similar projection images are indeed put into one class.

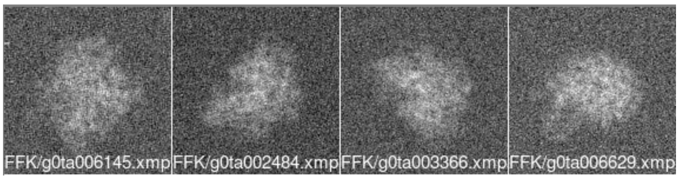


Fig 5: Images of Bacteriorhodopsin with white Gaussian noise at SNR=0.83.

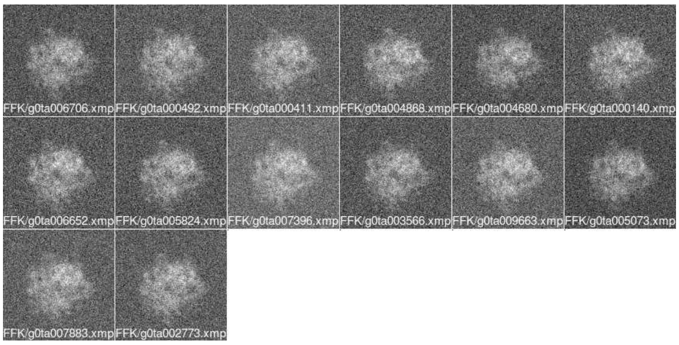


Fig 6: Images in one class from our classification result. We classify bacteriorhodopsin with white Gaussian noise at SNR=0.83.

To compare the results of CL2D and ours, we evaluate each class by computing the angles between the projection directions of any image assigned to that class and the representative of the class. Fig 8 shows the classification quality curves for classifying the projection images into 256 classes by CL2D without applying any filter and our method. The method yielding a narrower and higher probability density curve is better. To raise the classification accuracy, we expand the sizes of all projection images to 16 times of the original by adding zeros. Hence, the sizes of the projection images are increased to $(4*143)\times(4*143)$. The figure shows that, for this example, we obtain better classification result than CL2D. Fig 9 shows five class averages from our 256 classes. The execution time of CL2D running in parallel with 8 cores is about 273718 seconds. Our method using 8 cores takes about 486 seconds. Our method takes about 2580s using a single core. All the execution times presented in this section are wall time in second.

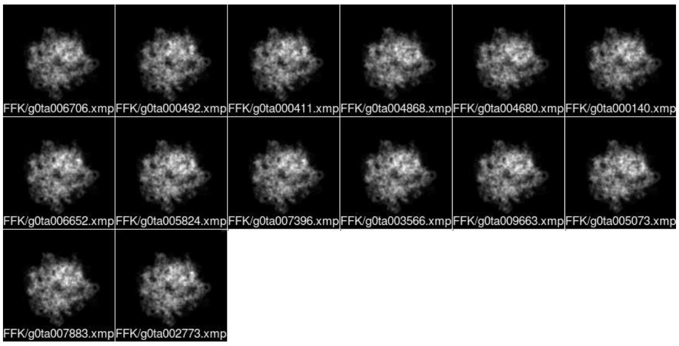


Fig 7: The projection images in the class shown in Fig 6 before adding the noise.

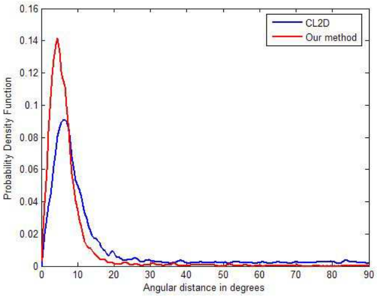


Fig 8: Clustering quality: probability density function estimation of the angular distance between each projection assigned to a class and its representative for CL2D and our method (SNR=0.83). Ideally, these functions should be concentrated at small angular distances, which means that the projection images assigned to a class have similar projection directions. The maximum angular distance is 90° because it is worst that two orthogonal views are assigned to the same class.

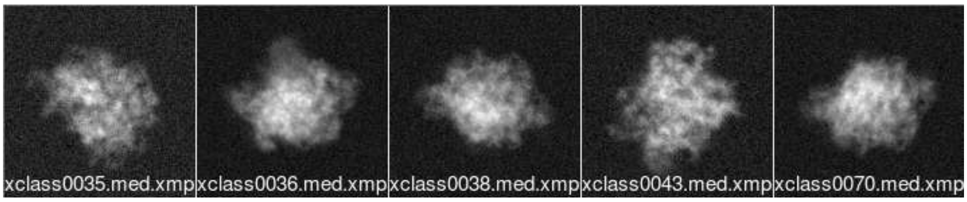


Fig 9: The class averages of five classes selected from 256 classes of bacteriorhodopsin classification (SNR=0.83).

## 4.2 Simulated Data: Bacteriorhodopsin with SNR=0.3

We add white Gaussian noise to the projection images of Bacteriorhodopsin with SNR=0.3 (see Fig 10). We classify the projection images into 256 classes and get the similar result to the classification of the images without adding noise. Fig 11 shows the images in one class from our classification result. Fig 12 shows all the images in Fig 11, but without adding noise. We can see that similar projection images are put into one class. Here the sizes of the projection images are increased to $(4 * 143) \times (4 * 143)$ in the same way as the experiment above. Fig 13 shows classification quality curves of CL2D and our method. In this classification we use Gauss low-pass filter for both CL2D and our algorithm. We can see that we get the similar classification result to CL2D. Fig 14 shows five class averages taken from our 256 classes. The execution time of CL2D running in parallel with 8 cores is about 306466s. Our method using 8 cores takes about 459s. Using a single core, our method takes about 2886s.

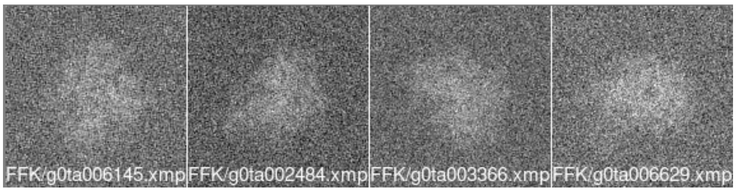


Fig 10: Bacteriorhodopsin projection images with SNR=0.3.

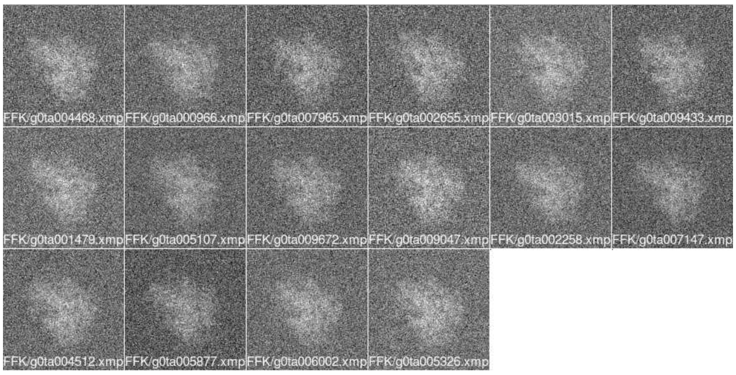


Fig 11: Images of one class from our classification result for bacteriorhodopsin projection images with SNR=0.3.

### 4.3 Simulated Data: E. coli Ribosome with SNR=0.03

We use a public data set of a simulated ribosome which is available at the Electron Microscopy Data Bank (http://www.ebi.ac.uk/pdbe/emdb/singleParticledir/SPIDER_FRANK_data (Baxter et al., 2009)). This data set contains 5000 projection images from random directions of a ribosome bound with three tRNAs at A, P and E sites. Fig 15 shows a few of the sample images from this data set with SNR=0.03. In this experiment, the same

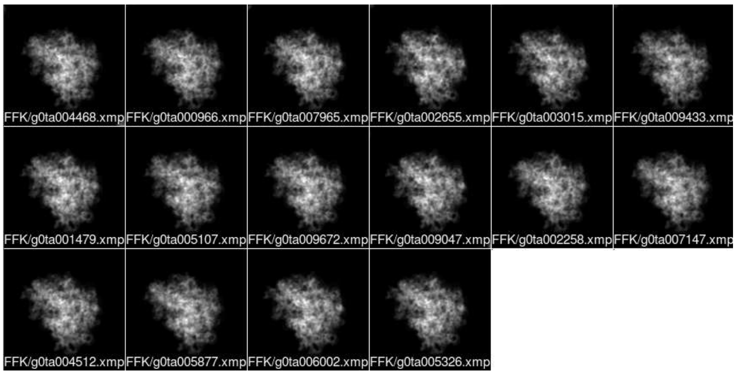


Fig 12: The projection images in Fig 11 before adding the noise.

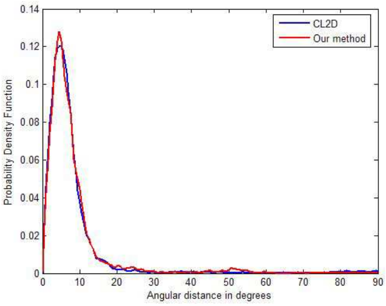


Fig 13: Clustering quality: probability density function estimation of the angular distance between each projection assigned to a class and its representative for CL2D and our method (SNR=0.3).

low-pass cut-off frequency components are used by CL2D, our direct classification and our classification after expanding the sizes of the projection images to $(4 * 130) \times (4 * 130)$. Fig 16 shows the classification quality curves for classifying the projection data into 256 classes by the three methods. It is easy to see that the direct application of our method yields a little better classification result than CL2D. When the image sizes of the projection images are increased, we get an even better result. Fig 17 shows five class averages from our 256 classes by the direct classification (without expanding). The execution time of CL2D running in parallel with 8 cores is about 264487s. Our direct method in 8 cores takes only about 48s. The classification after expanding the sizes takes 290s in 8 cores. Using a single core, our direct method takes 240s, and our method after expanding the sizes takes 1850s.

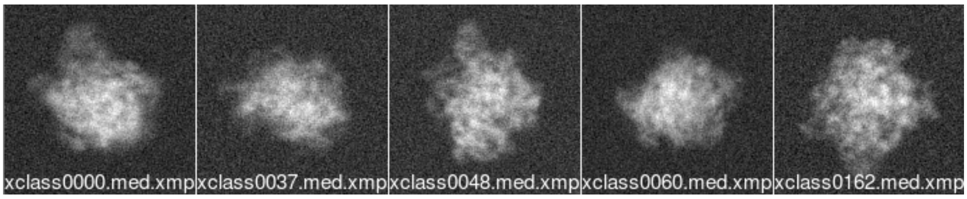


Fig 14: The class averages of five classes among 256 classes of bacteriorhodopsin classification (SNR=0.3).

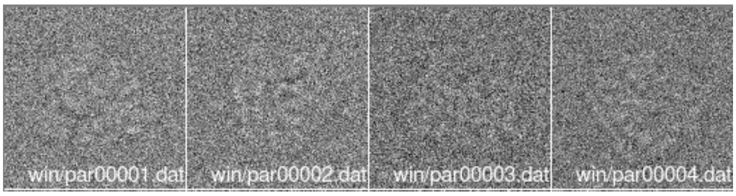


Fig 15: Images of E. coli ribosome, SNR=0.03.

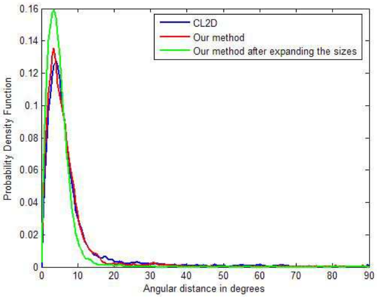


Fig 16: Clustering quality: probability density function estimation of the angular distance between each projection assigned to a class and its representative for CL2D and our method (SNR=0.03).

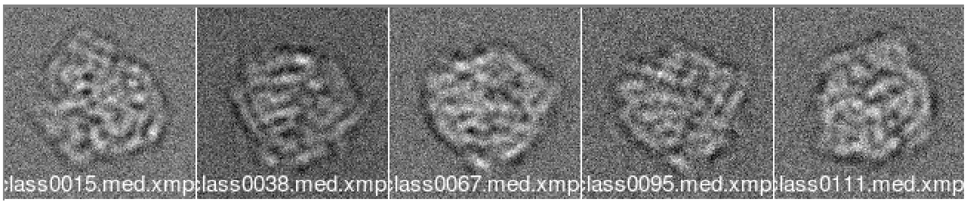


Fig 17: The class average of five classes among 256 classes of E. coli ribosom classification (SNR=0.03).

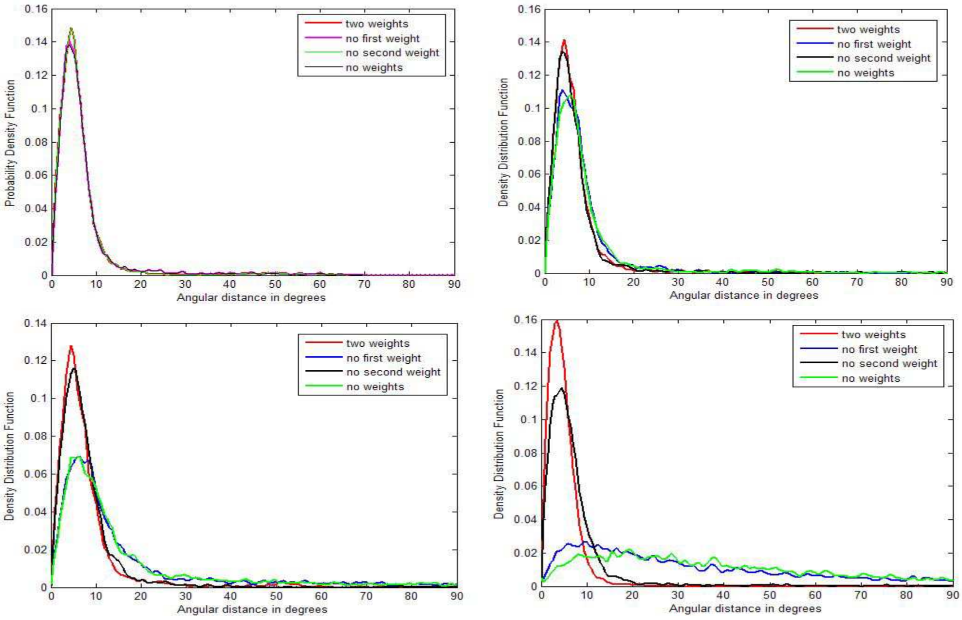


Fig 18: Probability density curves for classification using different weights for our method. Top left: Using data Bacteriorhodopsin without noise. Top right: Using data Bacteriorhodopsin with white Gaussian noise at SNR=0.83. All methods classify the images directly without filters. Bottom left: Using the data Bacteriorhodopsin with white Gaussian noise at SNR=0.3. Bottom right: Using the data E. coli ribosome, SNR=0.03, and expanding the sizes.

## 4.4 Comparative Results Using Different Weights in Our Method

In section 4.2, we have claimed that the weights in defining our invariant play an important role. In this subsection, we verify this by a few numerical examples. We test three data sets with adding difference levels of noises, to show how the noise affects the classification results. Fig 18 show the probability density curves for the classification for different noise level and different data sets. From these figures, we can draw the following conclusions:

1. The first weight is more important than the second one.
2. Without using the weights, the classification results are unacceptable for data sets with high noise.
3. If the data set is clean, all the results are acceptable, but the results with the weights are still better.

### 4.5 Comparative Results Using Different Invariants

We have mentioned that several invariants of translation and rotation have been proposed in the past decades. Each of them can be used to compute the similarities of different images to be classified. In this subsection, we present a few comparative results for using different invariants in our classification method. The first comparison is using the same data set with the same noise and using different invariants. Figure 19 shows the classification quality curves for classifying 5000 projection images into 256 classes. The probability density curves show that if the data set is clean (no noise), the invariant DSCF yields the best results, then our method follows. AFMT is the worst and DACF is the second worst one. If the data set has noise, our method yields the best results and all the other invariants give unacceptable results. This further illustrates that our method is much more robust.

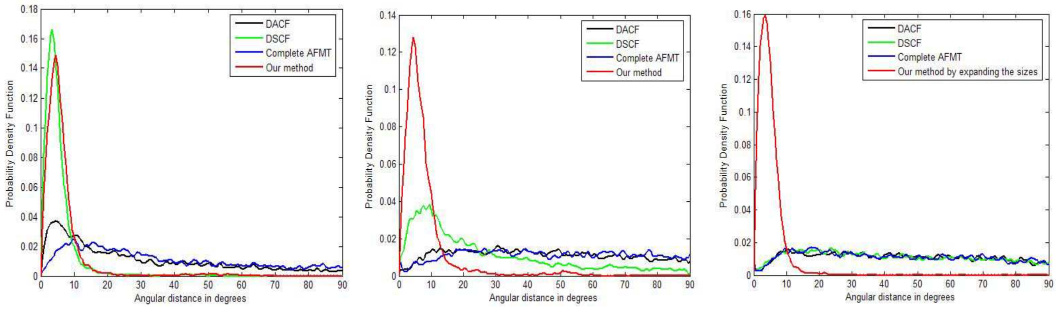


Fig 19: Probability density curves for classifying 5000 projection images using different invariants. Left: Using data Bacteriorhodopsin . Middle: Using data Bacteriorhodopsin with white Gaussian noise at SNR=0.3. Right: Using data E. coli ribosome with SNR=0.03.

In the second comparison, we compare the behaviors of each invariant with respect to the noises. The probability density curves in each figure of Fig 20 are for the same data set and same invariant but different noise levels. Again, the 5000 projection images of the data Bacteriorhodopsin are classified into 256 classes. The figures clearly show that our method is not sensitive to the noise and therefore it is robust. DSCF yields the second best results. The other two invariants do not lead to ideal results, especially for the highly noisy data.

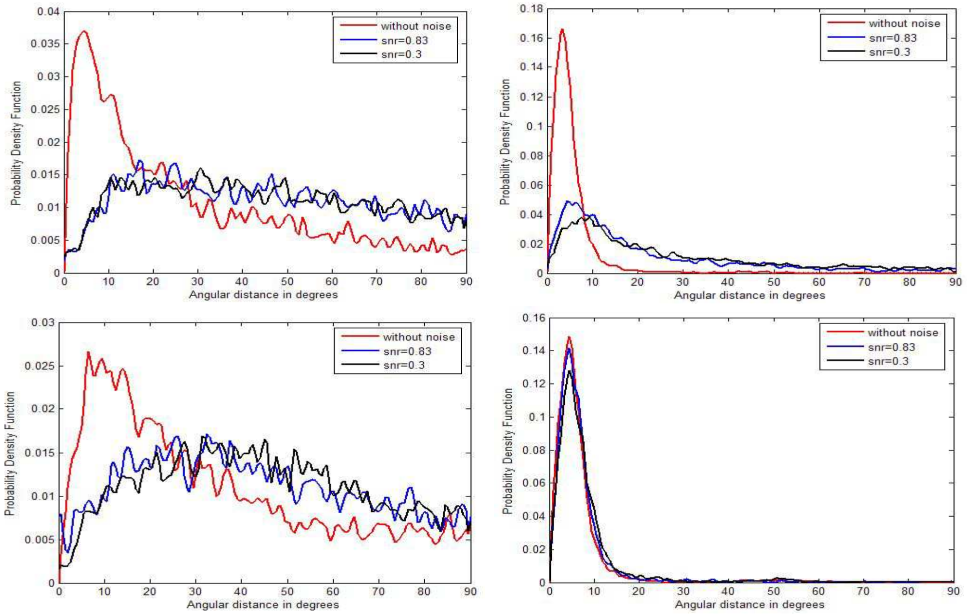


Fig 20: Probability density curves for classifying the projection images of Bacteriorhodopsin with different level of noises and using different invariants. Top left: Using DACF. Top right: Using DSCF. Bottom left: Using AFMT. Bottom right: Using our invariant.

**Remark 4.1.** *In defining our FTTR-invariant, we take module twice. This means that the phase information is lost. It is natural to think that using the complete invariant AFMT (no information loss) may be a more desirable choice. However, for solving the classification problem under the noisy circumstances, our invariant performs much better than the complete invariant. Hence, a problem that need to be further investigated is that why the complete invariant does not work well.*

**Remark 4.2.** *To fairly evaluate the classification methods, we use simulated data sets. Since the projection directions are known for the simulated data sets, we can compute exactly the quality curves for measuring the angle deviations. However, for the real data, since the projection directions are unknown, any evaluating criterion for the correctness of the classification is subjective. We therefore do not use real data sets.*

## 5 Conclusions

We have presented a fast and effective classification method for classifying electron microscopy images. Firstly, a weighted translation and rotation invariant based on the Fourier transform is introduced. Using this invariant, we avoid image alignment. A proper weight is selected bringing the success of our classification. Secondly, a proper cut-off frequency is chosen in the Fourier space, making our method antinoise and fast. Thirdly, a secondary-class associated with each of the classes is introduced, which speeds up greatly the classification-adjusting. Finally, the projection images are classified with an adaptive $\varepsilon$ so that the distribution of every class in the space is as close to uniform as possible. From our experiment results, we can see that the projection number in every class is nearly in the same order of magnitude. Furthermore, our classification method does not suffer from the *attraction problem* even for the data set with low SNR. The experimental results also show that our method yields equally good (sometimes better) results as C.O.S. Sorzano et al's with several hundred times faster.

We have also compared the performance of our invariant with other three existing invariants (DACF, DSCF and AFMT). Using our invariant yields much better classification results, especially for the noise data sets. Our implementation $C$ program is freely available from http://lsec.cc.ac.cn/~xuguo/misce.htm.

**Acknowledgments** The project was supported by NSFC under the grant (11101401), NSFC key project under the grant (10990013) and Funds for Creative Research Groups of China (grant No. 11021101). The authors acknowledge that the implementation of our algorithm has been put into an open-source image processing package for electron microscopy: XMIPP by Dr. C.O.S. Sorzano when he visited the authors during January 7-15, 2012.

## A Other Invariants of Translation and Rotation

In this appendix, we review a few existing translation and rotation invariants. All these invariants have been tested in our code.

**Auto-correlation function**. Auto-correlation functions (ACF) are well known and widely used in signal processing. Let $f$ be an integrable function in the $L^2$ sense. Then the one-dimensional ACF is defined as follows (see [12]):

$$ACF(r) = \int_{-\infty}^{\infty} f(x)f(x-r)dx.$$

It is easy to see that ACF is a translation invariant. Two-dimensional ACF is defined in the same way. To define a translation and rotation invariant for a two-dimensional image, the two-dimensional ACF is firstly computed and then the ACF is transformed into the polar coordinates. Finally, a one-dimensional ACF operation is applied to the angular coordinate, resulting a rotational invariant. Since auto-correlation functions are computed twice, the final result is called in [12] double auto-correlation function (DACF), which is a translation and rotational invariant.

**Self-correlation function**. Let $M$ be the module of the Fourier transform $F$ of the function $f$. The inverse Fourier transform of the module $M$ is named as self-correlation function (SCF) of $f$. Obviously, SCF is a translation invariant. To define a translation and rotation invariant for two-dimensional images, the two-dimensional SCF is firstly computed and then the SCF is transformed into the polar coordinates. Finally, a one-dimensional SCF operation is applied to the angular coordinate, resulting a rotational invariant. The final result is called double self-correlation function (DSCF), which is another translation and rotational invariant.

**Complete rotation invariant**. In the literatures [1, 4], the authors introduce a complete rotation invariant. By completeness, we mean that there is no information loss in the process of transformation. Therefore, the transformation is invertible. As a first step, the mass centers of the given images are set (by translation) to the origin of the coordinate system. Assume the centralized image, which is a translation invariant, is converted into the polar coordinate $f(r, \theta)$. Then compute the analytical Fourier-Mellin transform of $f(r, \theta)$ as follows.

$$\forall (k, v) \in \mathbb{Z} \times \mathbb{R},\ \mathcal{M}_{f_\sigma}(k, v) = \frac{1}{2\pi} \int_0^\infty \int_0^{2\pi} f(r, \theta) r^{\sigma - iv} e^{-ik\theta} d\theta \frac{dr}{r}.$$

From $\mathcal{M}_{f_\sigma}(k, v)$, a complete rotation invariant is defined as.

$$\forall (k, v) \in \mathbb{Z} \times \mathbb{R},\ \mathcal{I}_{f_\sigma}(k, v) = \mathcal{M}_{f_\sigma}(0, 0)^{\frac{-\sigma + iv}{\sigma}} e^{ik \arg(\mathcal{M}_{f_\sigma}(1,0))} \mathcal{M}_{f_\sigma}(k, v)).$$

Since $f(r, \theta)$ is a complete translation invariant, $\mathcal{I}_{f_\sigma}(k, v)$ is a complete translation and rotational invariant.

## References

[1] S. Derrode and F. Ghorbel. Robust and efficient Fourier-Mellin transform approximations for gray-level image reconstruction and complete invariant description. *Computer Vision and Image Understanding*, 83:57–78, 2001.

[2] J. Frank. *Three-Dimensional Electron Microscopy of Macromolecular Assemblies: Visualization of Biological Molecules in Their Native State.* Oxford, 2006.

[3] J. Frank, W. Goldfarb, D. Eisenberg, and T. S. Baker. Reconstruction of glutamine synthetase using computer averaging. *Ultramicroscopy*, 3:283–290, 1978.

[4] F. Ghorbel. A complete invariant description for gray-level images by the harmonic analysis approach. *Pattern Recognition Letters*, 15(1):1043–1051, 1994.

[5] S. Goh. The Mellin transformation: Theory and digital filter implementation. *Ph.D. dissertation, Purdue University, West Lafayette,I.N.*, 1985.

[6] R. Gonzalez and R. Woods. *Digital Image Processing.* Addison-Wesley Publishing Company, 2002.

[7] M. V. Heel and J. Frank. Use of multivariate statistics in analyzing the images of biological macromolecules. *Ultramicroscopy*, 6:187–194, 1981.

[8] M. V. Heel and M. Stoffler-Meilicke. Characteristic views of E.coli and B. staerothermophilus 30s ribosomal subunits in the electron microscope. *The EMBO Journal.*, 4:2389–2395, 1985.

[9] R. Kaur and MR. N. Aggarwal. Classification of knee MRI images. *Indian Journal of Computer Science and Engineering*, 2:356–363, 2011.

[10] F. Natterer and F. Wübbeling. *Mathematical Methods in Image Reconstruction.* SIAM, 2001.

[11] B. S. Reddy and B. N. Chatterji. An FFT-based technique for translation, rotation, and scale-invariant image registration. *Image Processing, IEEE Transactions on*, 5(8):1266 – 1271, 1996.

[12] M. Schatz and M. V. Heel. Invariant classification of molecular views in electron micrographs. *Ultramicroscopy*, 32:255–264, 1990.

[13] M. Schatz and M. V. Heel. Invariant recognition of molecular projections in vitreous ice preparations. *Ultramicroscopy*, 45:15–22, 1992.

[14] S. H. W. Scheres, M. Valle, R. Nunez, C. O. S. Sorzano, and R. Marabini et.al. Maximum-likelihood multi-reference refinement for electron microscopy images. *Journal of Molecular Biology*, 348:139–149, 2005.

[15] A. Singer, R. R. Coifman, F. J. Sigworth, D. W. Chester, and Y. Shkolnisky. Detecting consistent common lines in Cryo-EM by voting. *Journal of Structural Biology*, 169:312–322, 2010.

[16] C.O.S. Sorzano, J.R. Bilbao-Castro, Y. Shkolnisky, M. Alcorlo, and R. Melero et.al. A clustering approach to multireference alignment of single-particle projections in electron microscopy. *Journal of Structure Biology*, 171:197–206, 2010.

[17] Z. Yang and P. A. Penczek. Cryo-EM image alignment based on nonuniform fast fourier transform. *Ultramicroscopy*, 108:959–969, 2008.