Published in final edited form as:

J Struct Biol. 2009 May ; 166(2): 205–213. doi:10.1016/j.jsb.2009.01.004.

DoG Picker and TiltPicker: software tools to facilitate particle selection in single particle electron microscopy

N.R. Voss^{1,*}, C.K. Yoshioka^{1,*}, M. Radermacher², C.S. Potter¹, and B. Carragher¹

¹ National Resource for Automated Molecular Microscopy and Department of Cell Biology, The Scripps Research Institute, La Jolla, CA 92037, USA

² Department of Molecular Physiology and Biophysics, University of Vermont, Burlington, VT 05405, USA

Abstract

Solving the structure of macromolecular complexes using transmission electron microscopy can be an arduous task. Many of the steps in this process rely strongly on the aid of pre-existing structural knowledge, and are greatly complicated when this information is unavailable. Here we present two software tools meant to facilitate particle picking, an early stage in the single-particle processing of unknown macromolecules. The first tool, DoG Picker, is an efficient and reasonably general, particle picker based on the Difference of Gaussians (DoG) image transform. It can function alone, as a reference-free particle picker with the unique ability to sort particles based on size, or it can also be used as a way to bootstrap the creation of templates or training datasets for other particle pickers. The second tool is TiltPicker, an interactive graphical interface application designed to streamline the selection of particle pairs from tilted-pair datasets. In many respects, TiltPicker is a re-implementation of the SPIDER WEB tilted-particle picker, but built on modern computer frameworks making it easier to deploy and maintain. The TiltPicker program also includes several useful new features beyond those of its predecessor.

Keywords

TEM; cryo-electron microscopy; particle picking; random conical tilt; orthogonal tilt reconstruction

Introduction

The goal of particle picking is to extract individual particles from raw transmission electron microscope (TEM) images so that they can be classified, aligned, averaged, and back-projected into three-dimensional electron density maps. The problems faced in particle picking are many: the orientation dependent appearance of particles, the low quality of raw images (noise), the presence of background contamination (*e.g.* aggregation and ice crystals), the variable morphology of samples (*e.g.* blobs, rings, and filaments), and the large number of pixels that must be processed (efficiency).

Corresponding Author: Bridget Carragher, The Scripps Research Institute, 10550 North Torrey Pines Road, CB-129, La Jolla, CA 92037, tel: (858) 784-9070 fax: (858) 784-9090; email: bcarr@scripps.edu.

*These authors contributed equally to the manuscript.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

In the past, particle selection was often done manually, despite the large investment in time and the tedium such an undertaking entails. Despite the seeming banality of the task, the creation of an automated particle picker with the same flexibility and precision of a human is anything but trivial; particle picking is simply one of those tasks that are suited to the highly adaptive human visual system. People often develop effective, though somewhat enigmatic, criteria for selecting correct particles, and these criteria can be difficult, or impractical, to translate into code. As a result, days, even weeks, of researchers' time are spent manually picking modestly sized data sets, and the lack of an automated alternative is a significant bottleneck in single-particle processing. Manual picking also has several drawbacks, such as the introduction of bias and inconsistency due to fatigue and changes in user preference over time.

One of the major obstacles to improving the resolution of single-particle reconstructions is collecting and processing the exceptionally large data sets that are required. While researchers have achieved high resolutions ($<5\text{\AA}$) using manually selected datasets (on the order of 10,000 particles), these accomplishments came through a considerable expenditure of time, and exacting attention to the selection of particles (Zhang *et al.* 2008). Such results are also typically only achieved for well-characterized and well-behaved samples. For the vast majority of specimens, the careful manual selection of large datasets may not be practical in the face of problems like low symmetry, structural heterogeneity and other unknowns. Theoretical calculations (Henderson 1995; Glaeser 2004), as well as practical experience (Unwin 2005; Yu *et al.* 2008; Zhang *et al.* 2008), indicate that over one million asymmetric subunits are required to achieve resolutions high enough to trace the backbone of macromolecular complexes.

Many excellent automated particle-picking methods already exist that achieve impressive accuracy within specific domains (Zhu *et al.* 2004). The methods presented here are not the first, or likely last, attempts to automate this tedious process, but they do provide a highly practical, efficient, and fairly robust addition to the current arsenal of particle pickers.

Automated particle picking techniques

The most popular automated particle pickers use particle templates, usually an averaged, noise-free representation of the particle in a particular orientation. By cross-correlating the template with the raw TEM images, it is possible to identify the position of particles by finding peaks in the cross-correlation maps. The fast local correlation function (FLCF) is an addition to the common cross-correlation technique where the contribution of a circular mask is efficiently normalized out (Roseman 2004). The extra normalization reduces the number of false positives that can arise from edges and contamination. The SIGNATURE particle picking package implements a process in which the FLCF is further supplemented with an additional cross-correlation, the spectral correlation, that reinforces the distinctiveness of peaks corresponding to correct matches of the template (Chen & Grigorieff 2007).

Rather than using explicitly defined image templates, some particle-picking methods use mathematically pre-defined functions as "templates" for detecting particles. The particle picker we present in this manuscript is best placed in this category. In many cases, generic mathematical functions cannot, individually, discriminate particles very well, but several such functions can be combined through training to result in effective particle pickers (Mallick *et al.* 2004). Similar to the template-based methods above, this machine learning approach requires that the user supply a pre-existing dataset for it to work, which hinders their convenient use. Another method, in a somewhat similar vein, uses image filtering to find edges in images and then identifies circles and rectangles formed by these edges (Zhu *et al.* 2003). This method relies on robust edge detection, which is challenging for cryo-EM images, and only works best for samples with well-defined geometries.

A few approaches seek to find particles through more indirect means. An example of such an approach is to segment the background from the image and select the regions that remain as potential particles (Adiga *et al.* 2005). This requires significant amounts of image processing to result in a reliable segmentation, and substantial filtering to remove regions that would result in false positives. While this is the quintessential non-template based methodology, such an approach may still require many of the same parameters as other template-based approaches, such as the ideal shape and size of correct particles. A more recent entrant best fitted into this category uses the segmentation of particles from the background based on contours formed by convolution of the image with a Laplacian of Gaussian (LoG) function (Woolford *et al.* 2007). This particle picker is mathematically related to the particle picker we present here but, as discussed later, the implementation and philosophy are different.

It is difficult to outperform the accuracy of template-based approaches when the templates match the particles well, *e.g.* GroEL top and side views. As a result, template based methods are ideal for samples whose particles adopt a small number of preferred orientations (Zhu *et al.* 2004). On a randomly oriented sample, however, one has to decide between the use of many templates to more accurately represent the sample, or relaxing the threshold criteria, thereby introducing more false positives. The tradeoff is then between accuracy and efficiency. Template-based methods are also more computationally costly in situations where the templates possess little in-plane rotational symmetry. In these cases, it is necessary to do larger rotational searches, which if sampled too coarsely, also limit the accuracy advantages of the template-based approaches.

Due to vast differences between samples there is still, and likely will always be, a need for a variety of automated particle selection methods. We believe that our method, based on the Difference of Gaussian (DoG) scale-space construction (Lindeberg 1994), falls into a useful niche by providing an extremely fast, reasonably unbiased, and remarkably general approach, with the additional ability of being able to efficiently sort particles based on their size.

Particle Picking of Image Tilt Pairs—The main challenge in single-particle reconstructions is determining the orientation of each individual particle and then using that information to construct the three-dimensional structure of the macromolecule. The most reliable and simplest method is to compare the individual particles to projections from an initial 3D model. Only a few *ab initio* methods exist for obtaining an approximate structure of a macromolecule without the use of such an initial model. The most robust of these techniques require the collection of tilted pairs of images. Tilt pairs are two images, of the same set of particles, collected at two different tilts in the microscope. The geometry of these data collection methods provides a significant level of constraint that simplifies the determination of particle orientations and the identification of sample heterogeneity.

Random conical tilt (RCT) is a well-established method for obtaining an initial model from image tilt pairs (Radermacher *et al.* 1986). RCT relies on collecting a first image at a high tilt angle (typically 45–60°) followed by a second, untilted image. Since the radiation from the electron beam damages the sample during each image exposure, the first tilted image provides higher resolution information than the second. The untilted images are used to determine the 2D alignment and classification of each particle and this information, together with the tilt geometry, then determines the 3D orientation of the tilted counterparts so they can be reconstructed into a 3D map. A more recent method, orthogonal tilt reconstruction (OTR), uses a similar approach to RCT, but images are collected at tilt angles of –45° and +45° (Leschziner & Nogales 2006). This results in a 90° rotation between particle pairs and thus avoids problems associated with missing cones of data in RCT reconstructions.

We previously described an automated process for collecting tilted image data that allows us to generate a large number of tilt pair images efficiently (Yoshioka *et al.* 2007). This has created a bottleneck in the next stage of processing, which is the selection of particles from these tilted images. Unfortunately, particle selection in such datasets is more complicated than for standard single tilt datasets. In addition to the problems previously mentioned, particle selection of these datasets involves the accurate determination and bookkeeping of the relationship between particles in the image pairs.

To address this problem we have developed a particle picker with an interactive graphical interface that essentially provides a modern implementation of the SPIDER WEB tilt-picker software (Frank *et al.* 1996). The program, TiltPicker, is implemented in the python programming language and uses the wxWidgets GUI library for its interface front-end. This infrastructure makes it platform independent and readily extensible. We have also added a number of new features discussed further below, to help streamline the process of tilted pair picking.

Methods and Results

DoG Picker Design

Put simply, a Difference of Gaussian (DoG) map is the result of subtracting two Gaussian blurred versions of the same image. For review, the 1D Gaussian for any sigma, σ , is given by:

$$G(\sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{r^2}{2\sigma^2}} \quad (1)$$

The Gaussian blurred image, $I_G(\sigma)$, is obtained through convolution of $G(\sigma)$ with the original image $I()$:

$$I_G(\sigma) = I * G(\sigma) \quad (2)$$

The DoG image, for two different values of σ , is then:

$$I_{DoG}(\sigma_1, \sigma_2) = I_G(\sigma_1) - I_G(\sigma_2) \quad (3)$$

The purpose of the DoG is to efficiently approximate the Laplacian, or second derivative of the Gaussian, which has several useful properties regarding its response to edges and blobs in images. For several reasons, it is useful to define $\sigma_2 = k\sigma_1$, where k , the ratio between the two blurring radii, is called the k-factor. K-factors closer to 1.0 more accurately approximate the Laplacian of Gaussian (LoG), but as k becomes too close to 1.0, the result also approaches zero. In DoG Picker, the k-factor is determined based on the search range sampling when a range is given, but defaults to 1.1 when only a single target size is provided. In the human visual system, which is believed to behave at a low-level using a principle similar to DoG, the k-factor has been estimated as ~ 5 (Enroth-Cugell & Robson 1966; Young 1987; McMahon *et al.* 2004). Consequently, the features that make the Laplacian of Gaussian (LoG) useful are not overly affected by the k-factor used in the DoG approximation, though higher values of k will favor particles separated by a greater distances.

As shown in Figure 1, the DoG, and by extension, LoG, functions peak at the origin and are flanked by two zero-crossings and two negative lobes that asymptotically re-approach zero as the distance from the origin increases. During correlation, the functions respond most strongly

to image blobs that fall exactly within their zero crossings. Image perturbations that are larger begin to intrude into the negative lobes of the function and respond less strongly, and when large enough, produce an entirely neutral response since the integrated value of the function is zero. Since the zero-crossings of the LoG are located directly at the σ point, this makes it easy to tune the LoG to respond most strongly to blobs of specific sizes. The situation is slightly more complicated for the DoG, but it is still possible to determine two sigmas, σ_1 and $\sigma_2=k\sigma_1$, that maximizes the response to blobs of a specific size. By setting the DoG function equal to zero, it possible to find σ_1 and σ_2 in terms of the desired particle radius, r :

$$\sigma_1=r\sqrt{\frac{k^2-1}{2k^2\ln k}} \quad (4)$$

The DoG methodology can also be used to efficiently search over a range of sizes by repeatedly blurring the image by k multiples of σ , and subtracting adjacent images. The goal then, is to create a series of blurred images, $I_G(\sigma_1)$, $I_G(\sigma_2=k\sigma_1)$, $I_G(\sigma_3=k\sigma_2=k^2\sigma_1)$, ..., $I_G(\sigma_n=k^n\sigma_1)$ that can be easily subtracted to create multiple DoG maps with responses to objects of increasing size. This construction is known as a scale-space construction of an image (Lindeberg 1994).

A cost saving measure that can be used when searching over size ranges is to use previously blurred images in the generation of successive images. Since the result of a convolution of multiple Gaussians is another Gaussian, this means the size of σ can be kept smaller than would otherwise be needed. The relationship between cascaded Gaussian blurs is given by:

$$\sigma_2^2=\sigma_1^2+\sigma_D^2 \quad (5)$$

and since we want $\sigma_2=k\sigma_1$ we can then solve in terms of σ_D , the additional amount of blur needed to achieve $I_G(\sigma_2)$ given that we are starting with image $I_G(\sigma_1)$ rather than from $I()$. So we get:

$$\sigma_D=\sigma_1\sqrt{k^2-1} \quad (6)$$

And so the image $I_G(\sigma_2)$ can be computed as $I_G(\sigma_2)=I_G(\sigma_1)*G(\sigma_1\sqrt{k^2-1})$ rather than $I_G(\sigma_2)=I()*G(k\sigma_1)$. Under this formalism, we have two additional parameters the user might supply: the number of different sizes to pick, N , and the search range. An example of this process for $N=3$ and $r=65$ Å is given in Figure 2.

As a side note, it is interesting to point out that in Fourier space, the DoG (and LoG) act as band-pass filters whose support size is matched to the size of particle being searched for. This is shown in Figure 1.

Faster image generation

If image filtering is performed in Fourier space, then there is little advantage to using the DoG approximation rather than the actual LoG. The DoG does have several advantages, however, if the convolution is to be performed in real-space. In real-space the value of σ is directly related to the amount of processing required to perform the convolution. Since the Gaussian quickly approaches zero away from the origin, the size of the kernel can be greatly truncated with only a negligible effect on the accuracy of the convolution. In our case we use kernel sizes m , where $m=4\sigma$. This gets us from a time complexity of $O(n^3)$ to $O(nm^2)$. The next major advantage of the Gaussian kernel is that it is linearly separable, which means that the real-space 2D DoG

(or even 3D, 4D, etc.) can be calculated in separate 1D passes. This further reduces the cost of the convolution from $O(nm^2)$ to $O(nm)$. A final optimization is to use the property of cascading Gaussian blurs, as described in the previous section, to reduce the sizes of σ that are used, which has the practical effect of further reducing the kernel sizes m .

In real-world terms, the speed of the real-space DoG can be much faster than the Fourier-based solution, despite its overall inferior theoretical performance, $O(nm)$ vs. $O(nlgn)$. This has been born out in comparisons we have conducted (not shown), pitting the real-space approach against the Fourier-based approach using the FFTW library (Frigo & Johnson 2005). Certainly, once σ becomes large enough the Fourier-based approach will be faster, but this problem is generally avoidable, as such images can be binned by greater amounts, reducing both n and m .

Extracting peaks from the DoG map

To extract peaks from the DoG filtered images, we simply normalize the maps, and threshold areas that fall below a user supplied minimum threshold. This removes spurious peaks that may be caused by noise, background contamination, or incomplete particles. The connected components in the thresholded image are then found, and each one is assigned as a potential particle. In the cases where a size range is being searched, each DoG map is first processed separately to find potential particles for each size, and then later, overlapping candidates are reduced to the one with the strongest response. As an additional parameter, the user can supply an upper threshold that is a pixel value, above which, candidate particles are rejected. This maximal threshold is very effective at removing false positives caused by contamination or strong edges in the image.

Comparison of DoG Picker vs. template matching for GroEL

To compare the results of picking with DoG filter versus a normalized, standard template correlation (Roseman 2004), we collected and processed a data set using our test-bed macromolecule, GroEL. 530 images were collected using the Leginon automated data collection software (Suloway *et al.* 2005). From these images, DoG Picker picked 46,983 particles and the template correlation picked 33,650 particles, using top and side templates of the GroEL particle. These particles were then reconstructed using EMAN as described (Stagg *et al.* 2008). The resulting 3D structures of the DoG particles had resolutions of 7.75 Å for even-odd FSC $_{1/2}$ and 8.40 Å for R-measure (Sousa & Grigorieff 2007). The template correlated 3D structure was slightly better at resolutions of 7.16 Å for even-odd FSC $_{1/2}$ and 8.45 Å for R-measure (Figure 3C). The resolution values suggest that the DoG structure is virtually the same as the template-picked structure, but the DoG structure shows slightly less detail than the template-picked one (Figure 3, A and B).

Comparison of DoG Picker vs. template matching for 70S chloroplast ribosome

As a second test, we compared particle picking on 70S ribosomes. The structure of this sample was published earlier (Manuell *et al.* 2007), but works here as a good test dataset for addressing the differences between the pickers. The ribosome is a non-ideal macromolecule to pick with defined templates, because it is asymmetric and adopts many different orientations on the grid. It is nonetheless rather spherical, and thus, should be an almost ideal particle for DoG Picker. The 70S ribosome samples are also more structurally heterogeneous than GroEL, and are contaminated by disassociated ribosomal components. In particular, discrete 50S subunits in the sample are somewhat difficult to distinguish from complete 70S ribosomes. DoG Picker selected 26,415 particles from a set of 325 images, and template correlation picked 42,934 particles from the same set of images (using a single template). The resulting 3D structures of the DoG particles had resolutions of 13.60 Å for even-odd FSC $_{1/2}$ and 17.12 Å for R-measure (Sousa & Grigorieff 2007). The template correlated 3D structure was worse at resolutions of

13.73 Å for even-odd FSC_{1/2} and 17.88 Å for R-measure (Figure 4C). A key difference between the two structures is the improved appearance of the 30S small subunit in DoG Picker structure (Figure 4, A and B). Presumably, this is because DoG Picker did a better job of distinguishing whole 70S particles from partial 50S particles.

The TiltPicker program

While DoG Picker helps address the problem of picking unknown particles from images, the TiltPicker program provides a solution that is required when working with tilted pair datasets. Generally, the workflow in tilted particle picking is to load the two tilt pair images side-by-side so that the user can determine the transformation between the two images, associate the appropriate particles pairs, and pick those pairs (for example using the built-in DoG Picker). The job of TiltPicker then, is to make this process as efficient as possible by supplying automated and semi-automated tools to help perform these tasks.

As shown in Figure 5 (bottom row), the relationship between image pairs, and by extension, their particles, is based on a translation vector, two angles corresponding to the direction of the tilt axis in both images, and the level of image compression along the tilt axis as specified by the ‘perceived’ tilt angle, θ , in the microscope. The complete transformation is given by the formula:

$$\begin{bmatrix} x^{align} \\ y^{align} \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \cdot \begin{bmatrix} 1/\cos\theta & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix} \cdot \begin{bmatrix} x - x_2^{shift} \\ y - y_2^{shift} \end{bmatrix} + \begin{bmatrix} x_1^{shift} \\ y_1^{shift} \end{bmatrix} \quad (7)$$

The reverse operation (Figure 5, top row) is a similar formula:

$$\begin{bmatrix} x^{align} \\ y^{align} \end{bmatrix} = \begin{bmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \cdot \begin{bmatrix} x - x_1^{shift} \\ y - y_1^{shift} \end{bmatrix} + \begin{bmatrix} x_2^{shift} \\ y_2^{shift} \end{bmatrix} \quad (8)$$

The angles of the tilt axis, relative to the vertical axis, in the two images are ϕ and γ , respectively. The translation vector is calculated based on the same position in both images, relative to the origin of the images, $(x_1^{shift}, y_1^{shift})$ and $(x_2^{shift}, y_2^{shift})$. The tilt angle θ is the ‘apparent’ tilt angle difference between the two images used to model the image compression perpendicular to the tilt axis. It is important to note that the true tilt angle of the specimen stage in the microscope is not directly related to the perceived tilt angle. For example, a specimen imaged at tilts -45° and $+45^\circ$ will likely have a perceived tilt angle, θ of 0° , even though the true tilt angle difference is 90° . In the TiltPicker software, only the perceived tilt angle, θ is determined in transforming particles coordinates between images. Only upon 3D reconstruction must the true tilt angle be input as a constraint.

For the image transformation (Equations 7 and 8, above), the shift vector between the images is the most difficult parameter to calculate automatically. The initial shift is usually found by manually picking the same particle in both images. Attempts have been made to automate this first step, but it requires that the tilt axis and perceived tilt angle can be predicted effectively. The perceived tilt angle can be guessed based on the goniometer angle of the two images, but in practice, this estimate can be off by up to 10° , most commonly due to uneven carbon substrates. Feature-based matching, while appropriate in such unconstrained situations, is not reliable with high magnification low-dose TEM images because of the low signal to noise in the images. Cross-correlation can be used to solve this problem after high-pass filtering and a rotational search of the tilt axes, but this only succeeds when there is enough contrast in the

images. Since the initial shift determination failure rate is highly dependent on the sample, it was more efficient to implement a semi-automated method.

Determining the perceived tilt angle

The perceived tilt angle, θ , can be determined by comparing any three particles in the first image to three corresponding particles in the second image. The major caveat of this technique is that it is unable to determine the true tilt angle (required for later processing) and lacks the ability to distinguish between positive and negative tilt angles. The calculation follows a two-step process of calculating triangle areas and converting them to an angle. First, for every set of three particle pairs, a related triangle is formed in each image and the areas of the triangles are measured. The resulting perceived tilt angle, θ , is then determined by taking the inverse cosine of the ratio of the triangle areas from both images, A^{IMAGE1} , A^{IMAGE2} :

$$\theta = \arccos\left(\frac{A^{IMAGE1}}{A^{IMAGE2}}\right) \quad (9)$$

The image with the triangle having the largest area is the least tilted, usually untilted, image of the pair. Two features were introduced to offset the contribution of inaccurate triangles with small areas, *e.g.* particles are close to each other or lie along a straight line, and inaccuracies in the position of the associated particles. First, a hard minimum cutoff on the size of triangles is used (Figure 6B), and second, the calculated angle difference is determined from many triangles using a mean weighted by the area of the triangles, w_i (Equation 10, below). The resulting tilt angle is then biased toward the values given by triangles with large areas.

$$\theta = \frac{\sum w_i \arccos\left(\frac{A_i^{IMAGE1}}{A_i^{IMAGE2}}\right)}{\sum w_i} \quad (10)$$

After the perceived tilt angle difference and initial shift between the tilt pairs is known, the only remaining unknown variables are the tilt axis angles for each image. There are no direct methods for finding the tilt axes, so a simple least squares refinement of the RMSD between the calculated and actual particle centers (determined by Equations 7 and 8) is used (Figure 6C).

Additional tools

Several additional tools are provided to ease the burden on the user. First, the worst matched particle pairs, as determined by RMSD parameter fitting, are highlighted by yellow symbols to make them easy to identify and assess (Figure 6A). A ‘Clear Bad Picks’ button can quickly discard any erroneous particle pairs allowing the user to refine the parameters to improve the quality of the alignment (Figure 6A, bottom row). A second tool provided is the polygon removal tool. If there is large collection of erroneously picked particles, such as particles picked over carbon or from large aggregations, the user can draw a polygon around the offending region, in either image, and quickly clear all the corresponding particles from both images (Figure 6A, green polygon). Finally, the TiltPicker program offers the ability to mask both image pairs, based on the current transformation, so as to show only the actual areas of overlap between the two images (Figure 6A). If an image pair requires adjustment, this useful feature makes it visually easier to find and pick appropriate particles from the two images.

Implementation of DoG Picker

DoG Picker has been implemented within the TiltPicker program and significantly reduces the amount of time required to process image tilt pairs. The DoG Picker requires that the user input a threshold and the particle size (in pixels), which can be measured using the built-in ruler tool (Figure 6D). The tool then runs the DoG Picker on both images and only selects particles that are found in both images. The requirement that each particle must be found in both images serves as a double check and reduces the number of incorrectly picked objects. In this case, the DoG Picker is interactive and the parameters can be refined on the fly. After DoG picking particles, the tilt parameters can be refined using the existing pairs of matching particles and the user also has the option to manually add or delete any particles.

Compatibility with SPIDER

The TiltPicker program is designed to work with existing software tools for image processing. It can save to and read from four different file formats: XML, python pickle, text and SPIDER. The reading and writing of files has been modularized, so that the addition of any new preferred formats can easily be handled. Most importantly, a SPIDER format is provided; the SPIDER format saves the particle picks for both images and tilt parameters into one file, making it simpler to read and write. The file can easily be broken up into the three files required for most existing RCT or OTR reconstruction scripts.

Installation of the TiltPicker program and DoG Picker

The TiltPicker program is relatively easy to install; the only requirements are the scripting language, Python, and four Python modules: NumPy, SciPy (Jones *et al.* 2001), the python imaging library (PIL) and wxPython. All four python modules are free, open-source projects that are actively maintained and available for all major computer platforms.

DoG Picker is implemented from within TiltPicker, but is so simple in its design that it can be easily re-implemented quickly in any popular software package. For convenience, a command-line version of DoG Picker is provided with similar, but fewer dependencies as the TiltPicker program.

All software and source code is available for download at the website, <http://appion.org> including information on installing required dependencies.

Discussion

Quality of the DoG filter for picking particles

The human visual system is still unmatched in its ability to intuitively identify particles and no single automated method is yet general enough to meet the demands of all possible particle-picking situations. If the goal of particle picking is to minimize both the number of false positives (incorrectly picked particles) and negatives (missed particles), then at some point, any method will begin to trade one against the other, especially as subtleties like heterogeneous, incomplete, or damaged particles come into play. DoG Picker performs well at selecting particles whose appearance are “blob-like,” as shown in both the ribosome and GroEL data sets. This level of generality is a big advantage, but must be recognized as a limitation as well. The philosophy and implementation behind DoG Picker also makes it very efficient at splitting particles into different size groups, a unique feature among particle pickers without post-process sorting (White *et al.* 2004).

In this paper, we have shown that our DoG Picker can compete effectively with template-based methods, and actually outperform them within certain domains. A major weakness of the DoG blob-based detection is samples that are highly asymmetrical, hollow (*e.g.* empty viruses), or

worse, extremely disparate (*e.g.* large rings or long fibers). This means that it is difficult, if not impossible, to obtain a good response for them at any given size of constant radius. No picker is perfect (Zhu *et al.* 2004), but DoG Picker's simplicity means that in such circumstances, it might make an ideal preprocessing step for other particle picking methods. More specifically, we propose that the DoG responses might serve as better weak classifiers than the Haar functions used in the machine learning approaches of Mallick and colleagues (Mallick *et al.* 2004).

Differences to LoG picker

In principle, our approach is very similar to a recent technique that uses contours generated by the Laplacian of Gaussian (LoG) filter to separate particles from the background (Woolford *et al.* 2007). While their approach uses the same fundamental filtering technique as ours (DoG vs. LoG), the two methods differ in several key details of philosophy and implementation. The key philosophical difference is that their method seeks to use the LoG filter to find edges that best segment particles from the background. In contrast, our method uses the DoG filter to disclose the center of masses of the particles. While filters of smaller radii will more faithfully retain the shape of particles, this information is not usually crucial to particle picking, and is more susceptible to noise since it samples a smaller neighborhood of pixels. The number of pixels used in the DoG filter is larger, since the zero-crossings match the size of particles, and is thus more resistant to noise. The differences in implementation are technical; we chose to approximate the LoG by using the DoG, but these differences result in some practical advantages. By using the DoG filter, we gain efficiency with negligible impact on accuracy.

Comparison of DoG Picker and template matching for GroEL

Comparing the two GroEL structures from differently picked particles, the DoG Picker is of less quality, but still demonstrates that DoG Picker is a sufficiently accurate starting particle picker for a large class of commonly encountered particles. DoG picker is also much more time efficient than the template matching. Over all images the DoG picker averaged 6.7 ± 0.4 seconds per image, while template matching averaged 40.8 ± 0.4 seconds per image. A major contributor to the difference in resolution is that DoG Picker selects more false positives, as judged by eye, compared to template correlation, but a second major factor, is that DoG Picker also picked more oblique views of GroEL, compared to template correlation, which picked predominantly only top and side views. This is represented by the amount of particles assigned to each of the Euler angles (Figure 3, A and B, bottom right). For template picker, only the top and side views in the Euler distribution plots show high representation, whereas the DoG picked structure has a much more evenly distributed Euler angle distribution. That a more even coverage negatively impacted the resolution of the structure seems counter-intuitive, but is most likely due to incorrect assignments of the obliquely orientated particles. This is supported by the fact that the 5.4 Å structure of GroEL (Stagg *et al.*, 2008) also had predominantly only top and side views. These results demonstrate that DoG Picker competes well in a situation that should strongly favor template matching, and if efficiency were an important criterion, then it might prove the more attractive choice.

TiltPicker automation

In the TiltPicker program, all actions required from the user have been reduced to a series of mouse clicks for a small set of functions: (1) The user clicks the 'Find Initial Shift' button and the program provides an estimate. If the resulting shift is unsatisfactory the user can adjust it by manually selecting a matching pair of particles. (2) Having obtained the initial shift, the user can mask the images to display only their overlapping regions, which makes it considerably easier to visually identify matching particles in both images (Figure 6A). The user then manually picks a couple of particle pairs in the corners of both images. (3) Using

these particle pairs, the user requests an estimate of the tilt angle and tilt axis angles using the 'Find Theta' and 'Optimize Angles' buttons (Figure 6, B and C). (4) Using these estimates, the user can then request automatic picking for remaining particle pairs using the built-in DoG Picker function or use the results of other picking algorithms such as template matching (Figure 6D). (5) The user then has the option of using a number of manual tools to deselect large areas of the image that might contain bad particles (*e.g.* over the carbon, bad ice or bad stain) and manually add or remove picked particles. (6) The user then moves to the next image pair.

Despite a few remaining bottlenecks, particularly for samples embedded in vitreous ice, a fully automated version of TiltPicker program was implemented in our pipeline. The automated picker attempts to automate the TiltPicker steps listed above with a different method for estimating the initial shift and tilt axis angles. A rotational search for the tilt axis angles is conducted using the following steps: (1) Images are transformed by the perceived tilt angle, estimated from the microscope goniometer settings, and each tilt axis angle, (2) the transformed images are cross-correlated and (3) the tilt axis angles providing the largest cross-correlation peak are selected. In addition to obtaining the tilt axis angles, the cross-correlation peak location is used to calculate the location of the initial shift for the first pair of particles. At this point, if the tilt axis, tilt angle, and initial shift (*i.e.* first particle pair) are close to the true values then automated particle picking will proceed without error. In practice, most samples can be automatically picked using this method.

Although automated particle picking works under most conditions, when it does fail, it is usually related to problems obtaining the correct tilt parameters. An error can occur when a bad initial shift is found, the tilt axis angles are wrong or the true perceived tilt angle is different from the goniometer settings due to a slightly bent grid or uneven carbon substrates. Any of these errors can cause the particle matching algorithm to slip and align neighboring particles as particle pairs. These misaligned particles are then carried into the refinement algorithms resulting in larger errors and misaligned particles.

This misalignment is highly sample dependent and correlates with how closely the particles are packed in the images and also with the accuracy of the imported particle locations. As currently implemented, when the software fails to find the correct alignment parameters, it does not yet have a reliable metric for determining this failure. Because of these caveats, it is recommended that the user go through the image pairs and verify the particles picked between image pairs.

Conclusions

In this manuscript, we presented two software tools that assist in particle picking. The first tool, DoG Picker, is an efficient and reasonably general, particle picker based on the DoG image transform. It is a reference-free particle picker with the unique ability to sort particles based on size. The second tool is TiltPicker, an interactive graphical interface application designed to streamline the selection of particle pairs from tilted-pair datasets. The TiltPicker program includes several useful new features beyond those of its predecessor, SPIDER WEB. In addition, TiltPicker is built on modern computer frameworks making it easier to deploy and maintain. Together, these two particle pickers combine to streamline the process of creating initial models for macromolecules without pre-existing structural knowledge.

Acknowledgments

Funding for the work was provided by NIH grant RR23093. CY received addition funding through the ARCS foundation. This research was conducted at the National Resource for Automated Molecular Microscopy that is supported by the NIH through the National Center for Research Resources P41 program (RR17573). We would like to thank Edward Brignole and Albert Lueng for help with testing the TiltPicker program.

References

- Adiga U, Baxter WT, et al. Particle picking by segmentation: a comparative study with SPIDER-based manual particle picking. *J Struct Biol* 2005;152(3):211–20. [PubMed: 16330229]
- Chen JZ, Grigorieff N. SIGNATURE: a single-particle selection system for molecular electron microscopy. *J Struct Biol* 2007;157(1):168–73. [PubMed: 16870473]
- Enroth-Cugell C, Robson JG. The contrast sensitivity of retinal ganglion cells of the cat. *J Physiol* 1966;187(3):517–52. [PubMed: 16783910]
- Frank J, Radermacher M, et al. SPIDER and WEB: processing and visualization of images in 3D electron microscopy and related fields. *J Struct Biol* 1996;116(1):190–9. [PubMed: 8742743]
- Frigo M, Johnson SG. The Design and Implementation of FFTW3. *Proceedings of the IEEE* 2005;93(2):216–231.
- Glaeser RM. Historical background: Why is it important to improve automated particle selection methods? *J Struct Biol* 2004;145(1–2):15–8. [PubMed: 15065669]
- Henderson R. The potential and limitations of neutrons, electrons and X-rays for atomic resolution microscopy of unstained biological molecules. *Q Rev Biophys* 1995;28(2):171–93. [PubMed: 7568675]
- Jones E, Oliphant T, et al. SciPy: Open source scientific tools for Python. 2001
- Leschziner AE, Nogales E. The orthogonal tilt reconstruction method: an approach to generating single-class volumes with no missing cone for ab initio reconstruction of asymmetric particles. *J Struct Biol* 2006;153(3):284–99. [PubMed: 16431136]
- Lindeberg T. Scale-space theory: a basic tool for analysing structures at different scales. *J Appl Stat* 1994;21:2.
- Mallick SP, Zhu Y, et al. Detecting particles in cryo-EM micrographs using learned features. *J Struct Biol* 2004;145(1–2):52–62. [PubMed: 15065673]
- Manuell AL, Quispe J, et al. Structure of the chloroplast ribosome: novel domains for translation regulation. *PLoS Biol* 2007;5(8):e209. [PubMed: 17683199]
- McMahon MJ, Packer OS, et al. The classical receptive field surround of primate parasol ganglion cells is mediated primarily by a non-GABAergic pathway. *J Neurosci* 2004;24(15):3736–45. [PubMed: 15084653]
- Radermacher M, Wagenknecht T, et al. A new 3-D reconstruction scheme applied to the 50S ribosomal subunit of *E. coli*. *J Microsc* 1986;141(Pt 1):RP1–2. [PubMed: 3514918]
- Roseman AM. FindEM--a fast, efficient program for automatic selection of particles from electron micrographs. *J Struct Biol* 2004;145(1–2):91–9. [PubMed: 15065677]
- Sousa D, Grigorieff N. Ab initio resolution measurement for single particle structures. *J Struct Biol* 2007;157(1):201–10. [PubMed: 17029845]
- Stagg SM, Lander GC, et al. A test-bed for optimizing high-resolution single particle reconstructions. *J Struct Biol* 2008;163(1):29–39. [PubMed: 18534866]
- Suloway C, Pulokas J, et al. Automated molecular microscopy: the new Legimon system. *J Struct Biol* 2005;151(1):41–60. [PubMed: 15890530]
- Unwin N. Refined structure of the nicotinic acetylcholine receptor at 4Å resolution. *J Mol Biol* 2005;346(4):967–89. [PubMed: 15701510]
- White HE, Saibil HR, et al. Recognition and separation of single particles with size variation by statistical analysis of their images. *J Mol Biol* 2004;336(2):453–60. [PubMed: 14757057]
- Woolford D, Hankamer B, et al. The Laplacian of Gaussian and arbitrary z-crossings approach applied to automated single particle reconstruction. *J Struct Biol* 2007;159(1):122–34. [PubMed: 17490891]
- Yoshioka C, Pulokas J, et al. Automation of random conical tilt and orthogonal tilt data collection using feature-based correlation. *J Struct Biol* 2007;159(3):335–46. [PubMed: 17524663]
- Young RA. The Gaussian derivative model for spatial vision: I. Retinal mechanisms. *Spat Vis* 1987;24:273–93. [PubMed: 3154952]
- Yu X, Jin L, et al. 3.88 Å structure of cytoplasmic polyhedrosis virus by cryo-electron microscopy. *Nature* 2008;453(7193):415–9. [PubMed: 18449192]

- Zhang X, Settembre E, et al. Near-atomic resolution using electron cryomicroscopy and single-particle reconstruction. *Proc Natl Acad Sci U S A* 2008;105(6):1867–72. [PubMed: 18238898]
- Zhu Y, Carragher B, et al. Automatic particle selection: results of a comparative study. *J Struct Biol* 2004;145(1–2):3–14. [PubMed: 15065668]
- Zhu Y, Carragher B, et al. Automatic particle detection through efficient Hough transforms. *IEEE Trans Med Imaging* 2003;22(9):1053–62. [PubMed: 12956261]

Figure 1.

Shape of the DoG function. (A) Real space shape of the DoG functions for difference sigma values (B) The Fourier transform of the same DoG functions shown in panel A. Note how the DoG filter acts as a band-pass filter.

Figure 2.

Example of the DoG picker technique for picking particles of different sizes. The original image of GroEL particles (upper right) is first blurred by a sigma of 29.2 Å (upper left). Next the image is successively blurred three more times by the values at the far left to get final images blurred by the amount below the image (left column). Each successive image pair is then subtracted to get the DoG filtered map (right column) highlighting the particles of a different size for each DoG map. This example then picks particles of size 37.5 Å, 65.0 Å, and 112.7 Å.

Figure 3.

Three dimensional structures of GroEL resulting from particles picked using DoG picker (A-D, blue) and template correlation (E-H, pink). For both GroEL reconstructions there are top (A and E), intermediate (B and F) and side (C and G) views of the final structure. In addition, Euler angle distributions (D and H) display a histogram of the number of particles in a certain projection of the particle. The number of particles is represented by color, with green representing a high population and red a low population. (J) The even-odd FSC and Rmeasure resolution plots for the GroEL structures from DoG picker and template correlation. DoG Picker resulted in maps with resolutions of FSC_{1/2} at 7.75Å and Rmeasure at 8.40Å. Template correlation resulted in resolutions of FSC_{1/2} at 7.16Å and Rmeasure at 8.45Å.

Figure 4.

Three dimensional structures of chloroplast ribosome resulting from particles picked using DoG picker (A, blue) and template correlation (B, pink). (A) Stereo view of the chloroplast ribosome structure from the DoG picked particles. (B) Stereo view of the chloroplast ribosome structure from particles picked by template correlation. (C) The even-odd FSC and Rmeasure resolution plots for the chloroplast ribosome structures from DoG picker and template correlation. DoG picker resulted in maps with resolutions of FSC_{1/2} at 13.6 Å and Rmeasure at 17.1 Å. Template correlation resulted in resolutions of FSC_{1/2} at 13.7 Å and Rmeasure at 17.9 Å.

Figure 5.

Transformation operations for aligning image coordinate systems between an untilted and a tilted image. The original untilted image (image #1) is transformed into a tilted image by three operations (top row, from left to right). First, the original untilted image (image #1) is rotated counter-clockwise by its tilt axis angle (see image #2). Second, the rotated image (image #2) is compressed along the tilt axis, which is now vertical to the page (see image #3). Finally, the compressed image (image #3) is rotated clockwise by the tilt axis (see image #4). The reverse operation is used to transform the tilted image into the untilted image (bottom row, from right to left). The original tilted image (image #8) is rotated counter-clockwise (see image #7), expanded (see image #6), and then rotated back clockwise (see image #5). Each pair of images in a column have the same coordinate systems, but were transformed from different images. The red lines in images #5 and #4 show the overlapping region of the image when compared to images #1 and #8, respectively.

Figure 6.

The interactive TiltPicker display with pop-up dialogs. (A) The main window, both images are displayed side-by-side. The images are masked from the original square shape such that only the overlapping regions are shown. (B) Tilt angle calculator showing the calculated tilt angle, minimum triangle area and number of triangles used in the calculation. (C) The tilt parameters least-squares minimization refinement window. The six parameters are shown with their current values and the lock button (right column) to fix individual values during refinement. (D) The DoG Picker window with four options: particle size, threshold, particle contrast, and maximum number of peaks.