ULTRA-FAST REGISTRATION OF 2D ELECTRON MICROSCOPY IMAGES

Santiago García[(a)], Julio Kovacs [(b)] and Pablo Chacón [(a)]

[(a)] Department of Biological Physical Chemistry. Rocasolano Physical Chemistry Institute, CSIC, Serrano 119, Madrid 28006, Spain.

[(b)] QG Research, Los Angeles, CA

[(a)]pablo@chaconlab.org; sgarcia@iqfr.csic.es

ABSTRACT

Registration of noisy images is a crucial problem in multiple fields. In single-particle electron microscopy a huge number of two-dimensional (2D) image projections of biological molecules must be aligned multiple times to obtain a three-dimensional (3D) reconstruction. The 3D reconstruction quality is substantially determined by the accuracy and speed of the alignment method. Here, we implemented and validated a real-space correlation method based on Fourier-Bessel functions to perform the matching with a single 3D Fast Fourier transform. Our results show that we achieved faster and more accurate 2D alignments than state-of-the-art methods available in current image EM processing packages. The parallel-CUDA implementation achieves speedups ranging from 23 to 300 folds, depending on the angular sampling used. The achieved over-performance provides an excellent background for developing advanced single particle analysis algorithms towards higher resolutions. Besides, the method presented here is fully applicable to any 2D image rigid-body registration problem.

Keywords: image processing, 2D registration, electron microscopy, fast rotational matching, Bessel functions

1. INTRODUCTION

Single particle electron microscopy (EM) has become the most powerful technique in structural biology for studying large macromolecules and their assemblies. At near physiological conditions, the 3D structure of relevant biological complexes can be determined from sub-nanometer to even near atomic resolution (Frank, 2006). Using a small amount of purified sample, researchers process and average the collected 2D EM data into a 3D reconstruction. In the first step, numerous single molecule images are obtained from multiple EM measures. These images correspond to electronic density 2D projections of the molecule in different random orientations. Because such projections are extremely noisy images, they are aligned and classified by similarity to reduce the signal-to-noise ratio (SNR). In fact, single particle 3D reconstruction is an iterative alignment and classification procedure, where strong image averages produced by classification are used as reference images for the subsequent refinement steps. Therefore, the alignment, which ultimately determines the 3D reconstruction quality, is repeated multiple times. Such alignment typically maximizes a cross-correlation function (simple scalar product of the EM electron density values stored in the 2D images) between experimental noisy images and the reference images. Moreover, the computation time for 3D reconstruction increases with the number of images, becoming a bottleneck for high-resolution studies. Approximately 10^6 image projections are needed to target high-resolution. In this context, the complete process can take even days in a multiprocessor cluster. In summary, the alignment is a critical step that largely controls the efficiency and accuracy of the 3D EM reconstruction.

Current EM image processing packages use different alignment kernels to perform 2D registration efficiently, as described elsewhere (Joyeux and Penczek, 2002). The most popular approaches are the self-correlation method (SCF) and the resampling to polar coordinates (RPC) method. The latter resamples a fixed image into polar coordinate space with respect to several locations of the other image. By means of the Fourier convolution theorem, the rotational angle between projections is determined from a 1D fast Fourier Transform (FFT), whereas the two translational parameters (i.e., x and y shifts) are discretely scanned. In contrast, SCF is a Fourier method that decouples rotation and translation. The rotational angle is computed in the same way as in the RPC method, but using a mutual-correlation function instead the standard density cross-correlation. The translational parameters are obtained by a 2D FFT that greatly speeds up the alignment. Although SCF is much more efficient than RPC, it is less robust against noise (Cong et al., 2003; Joyeux and Penczek, 2002). To improve its accuracy, researchers have added a post local refinement step. This SCF plus refined protocol is the standard fastest alignment procedure. Alternatively, the fast rotational matching method (FRM2D) maintains the accuracy of RPC and is still competitive (2x slower) with SCF (Cong et al., 2003; Cong et al., 2005). In FRM2D, the alignment problem was recast into one translational and two rotational degrees of freedom. Instead of fixing one

image while rotating and translating the other, both

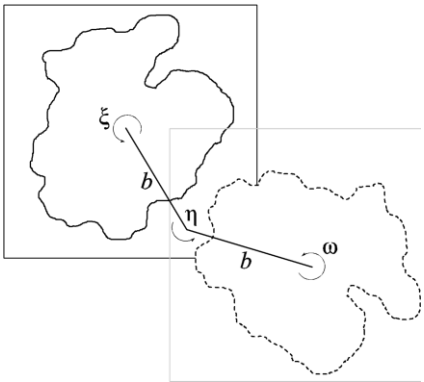

Figure 1. Matching setup of FBM method. The image on the left side is fixed, and the image on the right is moved by the rotations of ξ, η and ω to match the image on the left. *b* is a fixed value. In our implementation, *b* was fixed to 3.5 pixels.

images are rotated and one of them is translated along the axis formed between the images centers. This method accelerates the estimation of the two rotations by FFTs, while the remaining translational parameter is systematically explored (Cong et al., 2003; Cong et al., 2005). FRM2D has been implemented in the standard package EMAN and has been successfully used in challenging high-resolution 3D reconstructions (Cong et al., 2010). This method can also be considered as a 2D version of the 3D fast rotational matching procedure that we employed to predict protein interactions (Garzon et al., 2009) and to fit atomic structures into low-resolution EM density maps (Garzon et al., 2007).

More recently, Kovacs and collaborators (Kovacs et al., 2007) have outlined a new real-space correlation-based method to perform the 2D alignment step, known as the Fast Bessel Matching (FBM) method. In FBM, the matching problem is recast into three angular parameters that can be estimated by a single 3D Fourier transform. To speed up all the rotational degrees of freedom, the correlation function was expressed in terms of the Fourier-Bessel transform of the image projections. Theoretical estimates from FBM showed a much lower complexity than current alignment 2D algorithms (Kovacs et al., 2007). These results suggested FBM as the optimal real-space method for matching EM noisy images. However, this hypothesis has not yet been confirmed.

In this article, we present the first FBM implementation and its adaptation to High-Performance Computing (HPC) systems. The parallelization can be performed directly by farming the graphics processing unit (GPU) processors with different alignments. The main advantage is its relative low storage needs and its simplicity when implemented on the GPU. Next, we summarize the methodology employed. Then, we confirm its robustness and accuracy. Finally, we describe and test the GPU implementation.

2. METHODS

The 2D image alignment in single particle analysis should be considered as a template matching problem. The templates are either average images obtained using clustering or generated as 2D projections from 3D reference structure. The experimental EM projections are compared with the template images to find most similar alignment parameters. Although these registration parameters are typically defined by two translations and one rotation between a fixed and a moving image, here registration is defined by the three angles as depicted in Figure 1. With this set up, FBM allows the direct estimation of all cross-correlation values between two images as a function of the rotation angles ξ, ω and η. We provide a brief summary of FBM, which has been described in detail elsewhere (Kovacs et al., 2007). First, the template image Fourier-Bessel transform can be written as:

$$F_m(x) = \int_0^\infty \hat{f}_m(u) J_m(ux) u du \qquad (1)$$

where $\hat{f}_m(u)$ corresponds to the Fourier transform of a given image *f* sampled in polar coordinates, with *u* being the fixed radius. The term $J_m(ux)$ corresponds to a Bessel function of the first kind with order *m*. The Fourier-Bessel transforms $G_m(x)$ of the other image to be matched is obtained as in Eq. (1). Note that both Fourier-Bessel transforms are computed using *2B* angular samples, and only for $|m| \leq B$, where *B* is the bandwidth. To compute the correlations with respect to the three angular variables, we can use:

$$C(\xi,\eta,\omega) = 2\pi \sum_{m,h,m'} e^{i(m\xi + h(\eta+\epsilon) + m'\omega)} \times \int_0^\infty J_{m-h}(bx) J_{h-m'}(bx) \times F_m(x) \overline{G_{m'}(x)} x dx \qquad (2)$$

where ϵ is a shift angle added to η to avoid duplicated values in the solutions. In our implementation ϵ was fixed to π/(2*k*). Applying the following variable change in Eq. (2), we obtain:

$$h = h_1 + m' \quad m = m_1 + h_1 + m' \qquad \eta' = \xi + \eta \quad \omega' = \xi + \eta + \omega \qquad (3)$$

This equation yields a correlation $C'(\xi,\eta',\omega')$, whose Fourier transform is:

$$\hat{T}(m_1,h_1,m') = 2\pi e^{i(h_1+m')\epsilon} \times \int_0^\infty F_{m_1+h_1+m'}(x) \overline{G_{m'}(x)} \times J_{m_1}(bx) J_{h_1}(bx) x dx \qquad (4)$$

By taking the inverse Fourier Transform of this equation, we can compute directly the matching correlation values for all angular triplets. In other words, we can recover the best matching angle solutions by simply find maximal values from the correlation 3D matrix. Efficiency has been achieved for two reasons: *i)* all of the registration procedures are reduced to a single inverse FFT and *ii)* the integral Eq. (4) involved simple operations between pre-calculated terms. In fact, all image Fourier-Bessel transforms are calculated at once, as are the Bessel functions. FBM was implemented in both CPU and GPU following this simple pseudo-code:

```
Precompute all image Fourier-Bessel transforms.
For each EXP image{
    Load EXP image
    For each REF image{
        Load REF image
        Calculate correlation REF / EXP
        Search best correlations
        Store the corresponding angles
    }
    Search best matching for this EXP
    Store best matching
}
```

2.1. CUDA Implementation

The graphical device FBM procedure was performed in three steps: *i*) compute the integral of Eq. (4) *ii*) compute the inverse Fourier Transform *iii*) search the highest correlation and store its alignment values. To compute the 3D FFT, we employed NVIDIA's cuFFT library. In particular, we obtained the best performance employing cufftPlanMany interface with the native compatibility configuration. However, this configuration needs an extra temporary correlation matrix, doubling the memory required. Thus, the FFTs number and hence the 2D alignments computed in batch are limited by the available GPU memory. For example, in a GTX 470 card we were only able to process packages of 9280, 1120, and 320 FFTs, for bandwidths 64, 128 and 192, respectively. Note that bandwidths are multiple of 32 in agreement with NVIDIA's architecture guidelines. Constrained by FFT memory needs and by the bandwidth, we have designed two kernels: one to compute the integrals and other one to find the best matching results. The overall grid layout integral kernel is schematized in Figure 2. At the hierarchy top level, a grid is organized as a 2D array. The number of reference and experimental images computed in batch determined the dimensions of such 2D computational grid. At low level, the threads are organized in bandwidth size blocks to compute a single row of a given 3D matrix integral. Once the kernel has computed all the integrals, the inverse Fourier transforms are calculated by means of cuFFT. The resulting matrices containing all the correlation values are subsequently searched to find the highest match scores. The search is a quite serial process and a non-optimal task for the graphical device since correlation matrix transference to RAM is prohibitive. For the search kernel, each GPU block is devoted to process a single experimental image against all the references. Every thread in the block searches on its corresponding correlation matrix for the angular triplet (ξ, η and ω) with the highest value. This kernel ends transferring to the RAM the best reference registration and its parameters. After all the experimental images are processed, finally, on CPU the best matches are selected and stored. For this particular implementation, most of the computing time is spent the integral kernel (~75%) whereas FFT and correlation search consumed the remaining 10% and 15%, respectively.

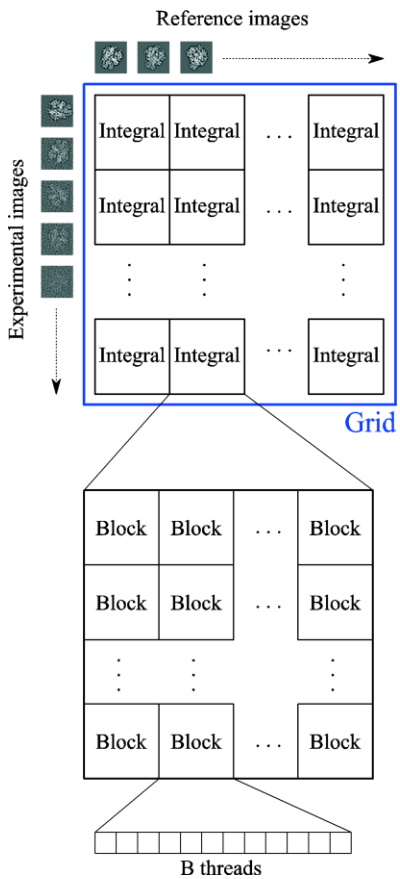

Figure 2. Grid layout of the main GPU kernel that computes the integral of Eq. (4). See details on the text.

2.2. Benchmark

To conduct the validation and comparison tests, we generated a matching benchmark from 2D projections of RNA polymerase II. This important macromolecule, which catalyzes the DNA transcription, has been characterized by EM in several conformational states (see, for example, Opalka et al., 2003). The simulated data have been obtained from the atomic structure (Protein Data Bank ID: 3M3Y) by using the single particle analysis software Xmipp (Marabini et al., 1996; Scheres et al., 2008). First, the electron density 3D atomic structure is projected into a 128 x 128 x 128 voxel density map with a 1.5 Å/voxel sampling rate by using the convert_pdb command. Then, a Gaussian low-pass filter was applied to simulate a 15 Å resolution

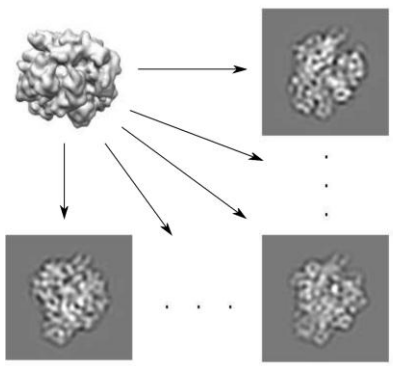

Figure 3. RNA polymerase EM map (grey surface) and some illustrative template projections (2D images).

map (fourier_filter command). From this simulated map, 80 random 2D projections of 128 x 128 pixels were created. Each projection corresponds to a given molecule orientation and, in principle, conforms to a different 2D shape (see Figure 3). To mimic real EM data, we also have simulated the microscope effect on these ideal projections using Xmipp phantom_simulate _microscope tool. This tool allowed us to add the contrast transfer function (CTF), such as defocus, astigmatism and lens aberration, and to simulate real noise with a given signal-to-noise ratio (SNR).

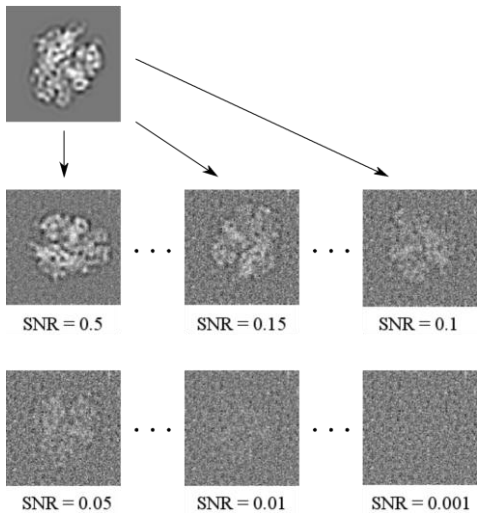

Figure 4. Illustrative examples of noise levels added to the random projections

Fifty different particle orientations were obtained by applying random rotations and translations to each reference image for generating a benchmark stack of 4000 image projections. The rotation shift was fixed to any angle between 0 and 360 degrees and, the translational shift was limited below 7 pixels. To simulate real data we also included noise to the images. We employed 19 different noise levels ranging from 0.5 to 0.001 of SNR, defined as $\sigma^2_{signal} / \sigma^2_{noise}$. Figure 4 illustrates the noise effect over the benchmark images. As can be observed, the macromolecule shape is only perceived at high SNR values. After applying all noise levels to every randomly oriented images, we had a total of 76000 experimental-like 2D projections.

The matching test will consist in recovering the original references of the whole experimental set by aligning them to the reference/template images. In principle, the highest cross-correlation values will correspond to the correct matches. The matching accuracy was measured by:

$$d\left|\sin(\Delta\phi/2)\right| + \sqrt{\Delta x^2 + \Delta y^2} \qquad (5)$$

where d is the particle diameter in pixels, $\Delta\phi$ is the relative angle misalignment, and x and y are the translational parameters (Joyeux and Penczek, 2002). The first term corresponds to the radial error, and the second corresponds to the translational error between reference and aligned images.

2.3. Technical details

These tests were conducted on a machine with an Intel i7 950 for the CPU and two NVIDIA GeForce GPUs, a GTX 470 and a GTX 680. The GTX 470 is a Fermi architecture card that has 448 cores along 14 multiprocessors, running at 1.22 GHz, and 1280 MB of RAM. The GTX 680 card has the latest Kepler architecture. It has 1536 cores along 8 multiprocessors running at a speed of 1.06 GHz and 2 GB of RAM available. To execute the FBM algorithm on the graphical device, we used CUDA 5.0 and NVIDIA driver version 302.59. The employ of the latest version of CUDA was mandatory to exploit the GTX 680 card capabilities.

3. RESULTS

3.1. CPU implementation

Because FBM has been never implemented, we first check its performance in a single CPU. To this end, we performed the matching test of the 4000 images against the 80 reference ones (see Methods) at three different bandwidths: 64, 128 and 192. These bandwidths correspond to an angular sampling of 2.8, 1.4 and 0.94 degrees, respectively. We also conducted the test at different noise levels to mimic the experimental conditions and test the method. The accuracy results of all 76000 matching experiments and their corresponding averages are shown in Figure 5. As it can be seen, the FBM method maintained subpixel accuracy (solid

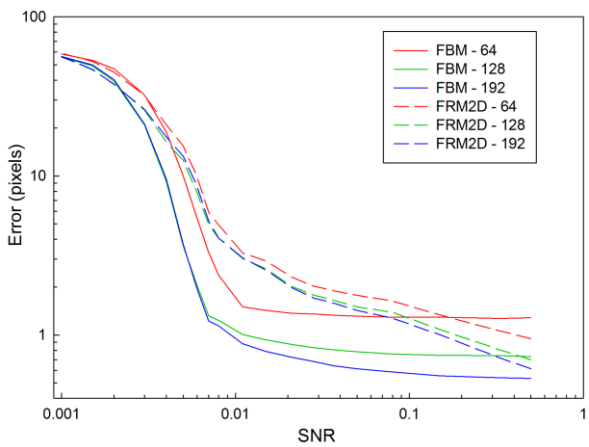

Figure 5. Average pixel error for FBM (solid lines) and FRM (dashed lines) for bandwidths 64 (red), 128 (green) and 192 (blue).

lines), even for SNR values close to 0.01. Around this region, alignments started to fail, and from this point, the error shown a fast accuracy loss. As expected, the accuracy improves as the angular sampling decreases (Figure 5). However, the differences between the 128 (green line) and 192 (blue line) bandwidths were not quite significant. The timing results are summarized in Table 1. FBM took less than 4 minutes with a bandwidth of 64 to match the 80 templates against 4000 experimental like images and less than an hour for 128. At the highest bandwidth FBM almost 5 hours were used to match the whole stack.

Table 1. Execution time in seconds for matching a stack of 4000 experimental test images against 80 reference images.

	CPU		FBM-GPU	
Bandwidth	FRM2D	FBM	GTX 470	GTX 680
64	2720	238	13.4	8.88
128	11293	3122	185	76
192	23939	16006	1014	443

For comparative purposes, we repeated the validation with FRM2D alignment method. This method exhibits better accuracy than the fastest SCF protocol at relatively low SNR ranges while keeps a good performance (Cong et al., 2005). We went a step further and, as suggested by Kovacs *et al.* (Kovacs et al., 2007), we optimized the FRM2D algorithm by reducing the bandwidth of an angular variable (Cong et al., 2003). This improvements result in a speed-up of one order of magnitude equaling the SCF performance. Therefore, in terms accuracy and efficiency, our FRM2D implementation is likely the best current reference method. In our tests, FBM clearly surpasses the accuracy of our optimized FRM2D version throughout the whole noise range (Figure 5). Even the FBM (solid lines) with the crudest sampling (64) is significantly more accurate than the FRM2D using thinner samplings (dashed lines) with SNR below 0.1. For example, at 128 the FBM error curve is always below that of FRM2D. The over-performance is more evident at the somewhat lower SNR levels. For example, at 0.01 of SNR, FBM matches with an error around 1 pixel, whereas FRM2D misses many alignments with an error larger than 3 pixels. From this SNR and under, the error recorded for both methods increases drastically, until both methods reach the maximum average error of 57 pixels at an SNR of 0.001. Fortunately, the typical EM experimental SNR ranges from 0.01 to 0.1.

The timing results are summarized in Table 1. For FRM2D, 45 minutes or up to 3 hours are needed for bandwidths of 64 and 128, respectively. On CPU, FBM provides speedups ranging from 1.5 to 11 times greater than FRM2D in the same conditions. These values are slightly better than the theoretical expectations, which estimated speedup gains around 2-5 fold (Kovacs et al., 2007). In summary, the obtained results confirm the superior accuracy and efficiency of FBM relative to the optimized FRM2D and by extension to current state of the art 2D alignment methods.

3.2. GPU implementation

Once the FBM was tested and its over-performance was demonstrated, we proceeded to implement a parallel version on the GPU by using CUDA. Parallelization was straightforward because the alignments of each reference image are independent tasks (see Method section for a detailed description).

To match each 4000 image stack, the CUDA version using GTX470 took 13.4 seconds, 3 minutes, or 17 minutes for bandwidths 64, 128 or 192, respectively. The executing times were between 66% and 40% smaller with newer and faster GTX 680 card. Thus, depending on the graphic device our CUDA implementation provides maximal speedups from 41x to 148x relative to the FBM-CPU version (Table 2). The speedup is not linear with the bandwidth. In the case of GTX 680 at 64, the card resources are not completely used and the speedup was limited to 27 fold. At bandwidth 192 the speedup decreases in both GPU cards. In this situation, the algorithm is likely to be saturating the device resources reducing the overall efficiency. However, in practical situations, angular samplings below 1º are not used because such accuracy is hardly achieved with noisy images. In summary, compared to the CPU, the GPU-FBM provides excellent performance. For example, it only took less than 10 minutes to match the whole 76.000 experimental image set, including all tested noise levels, with the fastest card at 128. More than 6.5 hours were needed for the CPU version in the same conditions.

If we now compare with currently used algorithms in CPU, such as FRM2D, using a GTX 470 card we found 203-, 61- and 23.6-fold speedups, for 64, 128 and 192 bandwidths, respectively. As expected, the speedups were substantially bigger for the GTX 680 ranging from 54 to 306. We did not implement the FRM2D GPU version because of its high memory requirements already pointed out in (Cong et al., 2003). Finally, it is important to mention that the CUDA version maintains the CPU accuracy in all cases tested.

Table 2. GPU speedups relative to CPU implementations

	GTX 470		GTX 680	
Bandwidth	FBM	FRM2D	FBM	FRM2D
64	17.7x	203.0x	26.8x	306.3x
128	17.8x	61x	41.1x	148.6x
192	15.8x	23.6x	36.1x	54.0x

4. Conclusions

In this paper, we implemented and validated a novel real-space and correlation-based 2D image registration algorithm. By means of Fourier-Bessel functions and a suitable recasting of the matching problem, we reduced the alignment process to calculate a single 3D FFT.

To verify the FBM robustness, we performed efficiency and accuracy tests with simulated RNA polymerase II images over a wide range of noise levels. The method maintains subpixel resolution at experimental-like noise levels. The GPU implementation boosts the efficiency between 16 and 41-fold with respect to the single CPU version. Moreover, compared with FRM2D, which is currently available on the *de facto* standard EM data processing package EMAN, FBM stands a significant improvement. In fact, for two different graphic devices we obtained speedups ranging from 23 to 300 folds relatively to our optimized FRM2D version. More importantly, FBM is considerably more accurate at experimental like noise conditions.

Based on the obtained results, our CUDA-FBM should be a sensible choice for image 2D alignment. It

will be particularly useful in the upcoming EM high-throughput scenario, where high-resolution structures and their huge number of projections have to be processed. In fact, we already had successful results with real experimental data using this novel approach. Our approach could complement recent CUDA developments in the EM image processing field (Castaño-Díez et al., 2010; Li et al., 2010; Schmeisser et al., 2009; Tagare et al., 2010) and other related Bessel based approximations for 3D reconstruction (Estrozi and Navaza, 2010).

Finally, our CUDA-FBM alignment kernel is a general method that can be useful in any application where the registration of multiple noisy images is required.

ACKNOWLEDGMENTS

This study was supported by BFU2009-09552, CAM-S2010/BMD-2353, CTQ2012-35873 and by the Human Frontier Science Program - RGP0039/2008. We thank P. Chys and JR Lopez-Blanco for their suggestions.

REFERENCES

Castaño-Díez, D., Scheffer, M., Al-Amoudi, A., Frangakis, A.S., 2010. *Alignator: A GPU powered software package for robust fiducial-less alignment of cryo tilt-series.* Journal of Structural Biology 170, 117-126.

Cong, Y., Kovacs, J.A., Wriggers, W., 2003. *2D fast rotational matching for image processing of biophysical data.* Journal of Structural Biology 144, 51-60.

Cong, Y., Jiang, W., Birmanns, S., Zhou, Z.H., Chiu, W., Wriggers, W., 2005. *Fast rotational matching of single-particle images.* Journal of Structural Biology 152, 104-112.

Cong, Y., Baker, M.L., Jakana, J., Woolford, D., Miller, E.J., Reissmann, S., Kumar, R.N., Redding-Johanson, A.M., Batth, T.S., Mukhopadhyay, A., Ludtke, S.J., Frydman, J., Chiu, W., 2010. *4.0-Å resolution cryo-EM structure of the mammalian chaperonin TRiC/CCT reveals its unique subunit arrangement.* Proceedings of the National Academy of Sciences of the United States of America 107, 4967-4972.

Estrozi, L.F., Navaza, J., 2010. *Ab initio high-resolution single-particle 3D reconstructions: The symmetry adapted functions way.* Journal of Structural Biology 172, 253-260.

Frank, J., 2006. *Three-Dimensional Electron Microscopy of Macromolecular Assemblies.* Oxford University Press, New York.

Garzon, J.I., Kovacs, J., Abagyan, R., Chacon, P., 2007. *ADP_EM: fast exhaustive multi-resolution docking for high-throughput coverage.* Bioinformatics 23, 427-433.

Garzon, J.I., Lopez-Blanco, J.R., Pons, C., Kovacs, J., Abagyan, R., Fernandez-Recio, J., Chacon, P., 2009. *FRODOCK: a new approach for fast rotational protein-protein docking.* Bioinformatics 25, 2544-2551.

Joyeux, L., Penczek, P.A., 2002. *Efficiency of 2D alignment methods.* Ultramicroscopy 92, 33-46.

Kovacs, J.A., Abagyan, R., Yeager, M., 2007. *Fast bessel matching.* Journal of Computational and Theoretical Nanoscience 4, 84-95.

Li, X., Grigorieff, N., Cheng, Y., 2010. GPU-enabled *FREALIGN: Accelerating single particle 3D reconstruction and refinement in Fourier space on graphics processors.* Journal of Structural Biology 172, 407-412.

Marabini, R., Masegosa, I.M., San Martín, M.C., Marco, S., Fernández, J.J., De La Fraga, L.G., Vaquerizo, C., Carazo, J.M., 1996. *Xmipp: An image processing package for electron microscopy.* Journal of Structural Biology 116, 237-240.

Opalka, N., Chlenov, M., Chacon, P., Rice, W.J., Wriggers, W., Darst, S.A., 2003. *Structure and function of the transcription elongation factor GreB bound to bacterial RNA polymerase.* Cell 114, 335-345.

Scheres, S.H.W., Núñez-Ramírez, R., Sorzano, C.O.S., Carazo, J.M., Marabini, R., 2008. *Image processing for electron microscopy single-particle analysis using XMIPP.* Nature Protocols 3, 977-990.

Schmeisser, M., Heisen, B.C., Luettich, M., Busche, B., Hauer, F., Koske, T., Knauber, K.H., Stark, H., 2009. *Parallel, distributed and GPU computing technologies in single-particle electron microscopy.* Acta Crystallographica Section D: Biological Crystallography 65, 659-671.

Tagare, H.D., Barthel, A., Sigworth, F.J., 2010. *An adaptive Expectation-Maximization algorithm with GPU implementation for electron cryo-microscopy.* Journal of Structural Biology 171, 256-265.