

# Tensorial template matching for fast cross-correlation with rotations and its application for tomography

Antonio Martinez-Sanchez<sup>1,\*</sup>, Ulrike Homberg<sup>2</sup>, José María Almira<sup>3</sup>, and Harold Phelippeau<sup>2</sup>

<sup>1</sup> Department of Information and Communications Engineering, University of Murcia, Campus de Espinardo, 30100, Murcia, Spain \*anmartinezs@um.es

<sup>2</sup> Materials and Structural Analysis Division, Advanced Technology, Thermo Fisher Scientific, Bordeaux, France

<sup>3</sup> Department of Engineering and Computers Technology, Applied Mathematics, University of Murcia, Campus de Espinardo, 30100, Murcia, Spain

**Abstract.** Object detection is a main task in computer vision. Template matching is the reference method for detecting objects with arbitrary templates. However, template matching computational complexity depends on the rotation accuracy, being a limiting factor for large 3D images (tomograms). Here, we implement a new algorithm called tensorial template matching, based on a mathematical framework that represents all rotations of a template with a tensor field. Contrary to standard template matching, the computational complexity of the presented algorithm is independent of the rotation accuracy. Using both, synthetic and real data from tomography, we demonstrate that tensorial template matching is much faster than template matching and has the potential to improve its accuracy.

**Keywords:** Template matching · Arbitrary object detection · Cartesian Tensors · Tomography

## 1 Introduction

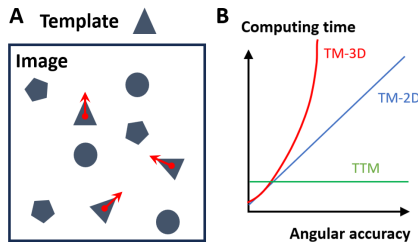
Template matching (TM) is a reference algorithm for arbitrary object detection in computer vision. In particular, TM is used for detecting instances of arbitrary patterns within images from just an input model of the pattern, the template. Contrarily to machine learning approaches, no prior training is required. TM is based on local normalized cross-correlation computation; thus, it presents a good performance for object detection when the instances of the template in the image are subjected only to translation and rotation transformations, no (or not significant) occlusions or any other deformation. Recently, deep learning (DL) approaches have been proposed for TM task improving its performance when previous conditions are not met on search images, but their outputs are biased by the training set used and none of them works for volumetric images (tomography) [14, 15, 35, 39].

As an example, in cryo-electron tomography (cryo-ET) [12, 19, 33], TM is used for macromolecular detection. Cryo-ET is an extension of cryo-electron microscopy (cryo-EM) for three-dimensional (3D) images, or tomograms, which has emerged as the unique technique able to generate 3D representations of the cellular context at sub-nanometer resolution [38]. TM is used to determine, within an image, the positions and rotations of instances of a structure (typically a macromolecule in cryo-ET) with respect to an input model, or template (see Fig. 1.A). Although multiple DL based particle picking algorithms are used for 2D cryo-EM single particle analysis (SPA) [1, 5, 23, 32, 40], its usage in cryo-ET remains very limited. Recently, DL approaches have been proposed to address macromolecular detection task for cryo-ET [22, 31, 37]. However, they present some limitations that make their usage restrained in practice. TM is required for preparing training datasets or processing datasets where the macromolecules are not visually recognizable. Moreover, DL methods cannot retrieve the rotations of the detected instances.

The main limiting factor for TM is its computational cost, especially when working with 3D images and using rotations. TM basic idea is to compute the cross-correlation by an inner product between the template and the image. Correlations are computed in Fourier domain to efficiently handle template translations [6, 29, 36]. However, this transformation must be repeated for every rotated version of the template. In the case of 2D images, the space of rotations can be identified with the unit circle  $\mathbb{S}^1$ , so its sampling leads to a linear computational complexity  $O(N) = O(360/\varepsilon)$ , for a given number of samples  $N$  determining the angular accuracy  $\frac{1}{\varepsilon}$ , implying that the maximal distance between the computed rotation and the exact one is smaller than  $\varepsilon$  degrees. Furthermore, this complexity increases notably in the 3D case, where the space of rotations is  $SO(3)$ , so that the complexity raises to  $O(N) = O((360/\varepsilon)^3)$ , which has a cubic dependency on angular accuracy  $\frac{1}{\varepsilon}$ . As an example, Reference [8] requires 45123 samples to achieve an angular precision of  $\varepsilon = 7^\circ$ .

In this paper, we propose and implement an alternative to classical TM algorithm, called tensorial template matching (TTM), with a computational complexity that is independent of angular accuracy (see Fig. 1.B). TTM is based on the recently developed mathematical theory described in [3]. To do so, TTM integrates into a tensor field, made of symmetric tensors, the information relative to the template over all rotations. This tensor field is computed only once per template. Therefore, TTM allows to find positions and rotations of template instances in an image with a reduced and fixed amount of correlations, just one per each linearly independent tensor component. We experimentally demonstrate that the new mathematical theory behind TTM is correct. Moreover, we also show that TTM is ready to substitute TM for some practical applications by processing real data.

This work uses tensors to encode information over all template rotations, an idea highly related with spherical harmonics. It has been shown that symmetric tensors can be represented using spherical harmonics and vice-versa [4]. However, for representing a function of  $SO(3)$ , the ordinary spherical harmonics are



**Fig. 1:** TM and its computational complexity. (A) TM consist in detecting positions, center localization, of the instances of an arbitrary input model, the template, in an image, and determining their rotations, or poses. (B) Computing time versus angular accuracy for TM in 2D images (blue) TM in 3D images (red) and the proposed TTM (green).

not enough. Hyperspherical harmonics [10] could be used, but these are fairly complex to handle compared to the simple Cartesian tensors used by the method presented here.

In terms of speeding up TM using cross-correlations, it is possible to reduce the area or volume over which the matching is performed by identifying regions of interest during a preprocessing step [6, 9]. However, this method risks missing matches if the preprocessing is not entirely accurate and depends on having a suitable preprocessing for a particular template. The method used in [9] finds likely candidates by preprocessing using rotation-invariant features. Afterwards, this method uses a similar basic approach for matching to TTM, considering that tensors and spherical harmonics are equivalent, but with an important difference: TTM does not need to explicitly consider multiple concentric shells, so it allows computing each correlation using the fast Fourier transform.

Steerable filters [13] have a similar efficient computational scheme compared to TTM, since for a specific kernel, or template, they use a linear combination of few basis filters to evaluate the convolution at any orientation of the kernel. The design of steerable filters has traditionally been limited to simple kernels [25]. Recently, a method has been proposed to generate steerable filters for any arbitrary kernel [11]. However, it only offers a practical solution for 2D images. Conversely, our approach solves the problem of TM computational complexity for 3D images.

## 2 Template Matching

Here, we introduce TM as it serves as the foundation for the subsequent description of TTM. TM relies on computing the local normalized cross-correlation (LNCC) function of the image,  $f$ , with respect to the template,  $t$ , for every voxel  $x \in \mathbb{R}^d$  and considering both images elements of  $L^2(\mathbb{R}^d)$ , where  $d$  is the dimension, e.g. in 3D  $d = 3$ . The LNCC can also be seen as a normalized inner product between  $f$  and  $t$

$$c(x, R) = w(x)(f \star t'_R)(x), \quad (1)$$

being

$$w(x) = \frac{1}{\sqrt{\langle S(\tau_x(f))^2, \mathbf{1} \rangle - \frac{\langle S(\tau_x(f)), \mathbf{1} \rangle^2}{M}}} \quad (2)$$

the normalization factor for the image. Here,  $\tau_x(f)(z) = f(z+x)$  represents the image translation at position  $x$  to fit the template center, and  $S$  is an operator implemented as

$$S(f) = m(h * f), \quad (3)$$

where the mask  $m$  equals 1 within a certain radius around the center and 0 outside a slightly larger radius; in-between these radii the mask interpolates between 0 and 1.  $M$  is the sum of voxels in  $m$ . Operator  $S$  also includes the separable filter  $h$  whose 1D  $z$ -transform is  $1 + a(z + z^{-1} - 2)$ . For the filter to be a low-pass filter and positive semidefinite, we should have  $0 < a < 1/4$ ;  $a = 1/5$  was chosen for its near isotropic behavior.

The template is zero-normalized by subtracting its mean  $\mu = (S(t) \cdot m)/M$  before the scaling

$$t' = \frac{m(S(t) - \mu)}{\sqrt{\langle S(t)^2, \mathbf{1} \rangle - \frac{\langle S(t), \mathbf{1} \rangle^2}{M}}}. \quad (4)$$

The cross-correlation between the image and the template is mathematically expressed as

$$(f \star t)(x) = \langle \tau_x(f), t \rangle = \int_{\mathbb{R}^d} f(z+x)t(z)dz. \quad (5)$$

In Eq. (1) the template is rotated by  $R$ . The whole space of rotations is sampled to generate the final cross-correlation field. The output is a scalar map  $c_{R_{opt}} : \mathbb{R}^d \rightarrow \mathbb{R}$  with the highest correlation values for each voxel

$$c_{R_{opt}}(x) = c(x, R_{opt}(x)), \quad (6)$$

$R_{opt}(x)$  is the optimal rotation at a given voxel

$$R_{opt}(x) = \arg \max_R \{c(x, R)\}. \quad (7)$$

The highest peaks of  $c_{R_{opt}}$  determine the location of different instances of the template in the tomogram. This also means  $R_{opt}$  on peaks estimates instances rotations. Accordingly, the density used for sampling the rotations space determines rotation accuracy as well as the detection performance.

### 3 Tensorial Template Matching

In contrast to TM, tensorial template matching, TTM, introduces the concept of tensorial correlation:



$$C_n(x) = w(x)(f \star T(t))(x), \quad (8)$$

being  $T(t)$  the tensorial template defined as

$$T(t) = \int_{SO(d)} R^{\odot n} S(t')_R dR, \quad (9)$$

where  $R^{\odot n}$  is a symmetric tensor of degree- $n$  constructed as the  $n$  tensor power for rotation  $R$ . In 3D,  $R \in SO(3)$ , then rotations can be expressed as unit quaternions  $q$  with  $d' = 4$ . The Algorithm 1 describes how to construct the tensor. This algorithm computes numerically Eq. (9), which requires to sample uniformly  $SO(3)$  space [2].

---

**Algorithm 1** Tensor template generation

---

**Require:**  $t$ : template,  $m$ : mask,  $Q$ : list of unit quaternions uniformly sampling  $SO(3)$

**Ensure:**  $T$ : tensorial template

```

    Compute  $t'$  ▷ Template normalization
     $I \leftarrow$  indices independent components degree- $n$  tensor
    for  $i \in I$  do
        Initialize  $T_i$  to a blank array of the size of  $t$ .
    end for
    for  $q \in Q$  do ▷ Integration in  $SO(3)$ 
         $t_{temp} \leftarrow t'_q$  ▷ Rotate  $t'$  according to  $q$ 
        for  $i \in I$  do
             $k \leftarrow$  generate component  $i$  of  $q^{\odot n}$ 
             $T_i \leftarrow T_i + kt_{temp}$ 
        end for
    end for

```

---

In Eq. (8),  $C_n : \mathbb{R}^d \rightarrow S^n(\mathbb{R}^{d'})$ ,  $x \mapsto C_n(x)$ , is a tensorial field, where  $S^n(\mathbb{R}^{d'})$  stands for the space of symmetric tensors of degree- $n$  and dimension  $d'$ . The key of TTM is that  $C_n(x)$  does not depend on rotations  $R$ , the information for the template over all rotations is integrated in the so-called tensorial template,  $T(t)$ , defined in Eq. (9). It is important to remark that the numerical computation of the tensorial template requires to sample the whole space of rotations,  $SO(3)$  in 3D, but it is only computed once for each template. Afterward, we can use this tensorial template to compute the tensorial field,  $C_n(x)$ , for any given image  $f$ . The Algorithm 2 describes how to generate the tensorial field. Notice that correlations are computed in Fourier domain ( $\mathcal{F}$ ).

The tensorial correlation requires as many correlations as linearly independent components of a degree- $n$  tensor, specifically  $\binom{n+d'-1}{n}$ . In principle, the higher tensor degree- $n$  the better  $T(t)$  integrates  $t$  information, however the theoretical speed-up of computing  $C_n(x)$  over  $c_{R_{opt}}(x)$  is reduced. Here, we work with tensors of degree- $n = 4$  for processing 3D images,  $d' = 4$ , therefore  $C_n(x)$

**Algorithm 2** Tensorial field computation

---

**Require:**  $f$ : image,  $T$ : tensorial template  
**Ensure:**  $C_n$ : tensorial field

```

  Compute  $w$  ▷ Local normalization of  $f$ 
   $I \leftarrow$  indices independent components degree- $n$  tensor
  for  $i \in I$  do
     $C_i \leftarrow w\mathcal{F}^{-1} \{\mathcal{F}(S(f))\mathcal{F}(T_i)^*\}$ 
  end for

```

---

computation will require  $\binom{7}{4} = 35$  correlations against the thousands typically used to compute  $c_{R_{opt}}(x)$ .

Now the challenges are the determination of the optimal rotation for a given position and the localization of the template instances in the image.

### 3.1 Optimal rotation

The main theorem in [3] states that, if there is a match between  $f$  and  $t_{R_{opt}}$  at  $x$ , then, for  $n$  even,  $C_n(x) \cdot R^{\odot n}$  gets a global maximum when  $R = R_{opt}$ . In addition, it has also been proven that finding this global maximum corresponds with finding the dominant eigenvalue-eigenvector pair of  $C_n(x)$ . Therefore, determining the optimal rotation for a given voxel,  $x$ , can be accomplished using the shifted symmetric higher-order power method (SS-HOPM) introduced in [27]. Initialization is performed using iterative eigendecomposition, as suggested by [26], in combination with simply trying 1000 uniformly distributed quaternions. The latter is especially useful when templates contain symmetries because the iterative eigendecomposition used to fail. The shift used in the SS-HOPM is determined using an eigendecomposition of the square matrix unfolding of the tensor, inspired by the observation that the algorithm converges whenever this matrix is symmetric positive semidefinite [26]. This shift appears to be more conservative than the one proposed in [34], while requiring less effort than the adaptive SS-HOPM proposed by the same authors [27].

### 3.2 Instance positions

Computing the optimal rotation only at the matching positions is crucial to maximize the computational advantage of TTM over TM. In this work, we propose to convert the tensorial field

$$C_n : \mathbb{R}^d \rightarrow S^n(\mathbb{R}^{d'}), \quad x \mapsto C_n(x)$$

into a scalar field

$$\hat{c}_n : \mathbb{R}^d \rightarrow \mathbb{R}, \quad x \mapsto \hat{c}_n(x)$$

with the following properties:

1. The computation of  $\hat{c}_n(x)$  is not expensive.

2. Matching positions correspond to local maxima of  $\hat{c}_n$ , so they can be efficiently determined by a peak detector.

Therefore, the important step is to define a candidate for  $\hat{c}_n(x)$  whose local maxima approximate as close as possible the global maxima over rotations of  $C_n(x) \cdot R^{\odot n}$ , meanwhile their computation is not expensive.

On one hand, let us take into account some properties of tensors. As described in [3], for any symmetric tensor  $T$ , it is known that:

$$\|T\|_\sigma = \max_{\|Q\|=1} |\langle T, Q^{\odot n} \rangle| = \max_{\|Q\|=1} |T \cdot Q^{\odot n}|,$$

where  $\|T\|_\sigma$  denotes the spectral norm of  $T$ . Thus,

$$\hat{c}_n^*(x) = \|C_n(x)\|_\sigma$$

satisfies the second condition but, unfortunately, it fails with the first one since the computation of the spectral norm of a tensor is NP-hard [18]. Indeed, in [21] is shown that if all but two of the  $n$  dimensions are fixed, then the spectral norm of an order  $n$  tensor can be computed in polynomial time, while it becomes NP-hard if three or more dimensions are taken as input parameters. In any case, computing  $\|C_n(x)\|_\sigma$  for all positions  $x$  with a given (small) precision  $\delta > 0$  is too expensive for our purposes.

On the other hand, in [3] the Frobenius norm  $\|T\|_F$  of a tensor was proposed as an alternative to the spectral norm. Here we show that it can be used as an excellent proxy for finding the spatial locations of instances. Indeed, we know that  $C_n(x) \in S^n(\mathbb{R}^{d'})$  and (see, e.g., [7], [28])

$$\|T\|_\sigma \geq \|T\|_F \frac{1}{2^{n-1}}$$

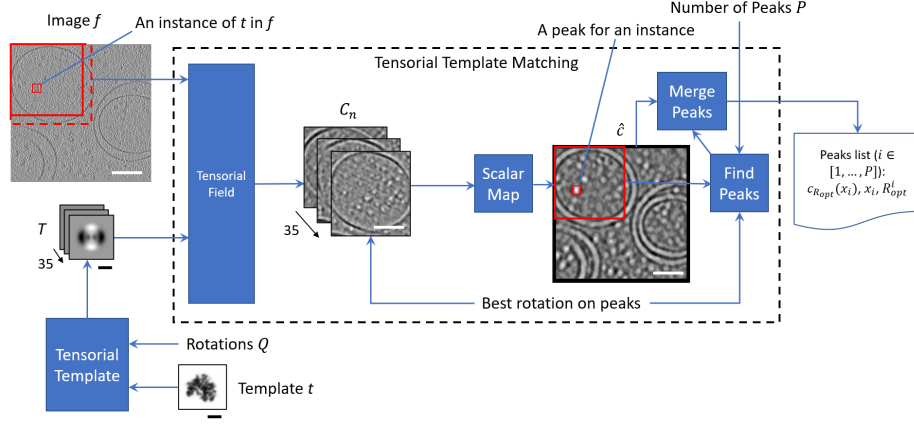
Thus, if  $\|T\|_F$  is large, the spectral norm of  $T$  is also large. It follows that a good candidate for the scalar field  $\hat{c}_n$  which satisfies the first condition and approximates the second is

$$\hat{c}_n(x) = \|C_n(x)\|_F.$$

For each position identified as a potential peak, the SS-HOPM algorithm is then used to find the exact dominant eigenvalue (and its corresponding eigenvector).

### 3.3 Positions refinement

As mentioned above and shown experimentally in Sec. 4, the Frobenius norm is a fast proxy for finding positions, but the exact match may be shifted a bit. When positions are shifted, the conditions for finding the optimal rotation are not met (see Sec. 3.1), so rotation determination is considerably degraded. Consequently, we propose here a heuristic to ensure that this potential shifting is corrected,



**Fig. 2:** TTM implementation scheme. Images represents 2D slices of tomograms, the input tomogram was taken from SHREC dataset [16], and the template was generated from PDB-5MRC atomic model. Black scale bars 200 Å, white bars 2000 Å.

which is so called TTM-refined (TTM-ref). First, for each peak found, we define a sphere around it with a radius of  $r_s$  voxels. Second, we iterate over these voxels and compute the optimal rotation as described in Sec. 3.1. Finally, the LNCC at just the optimal rotation,  $c(x, R_{opt})$ , is computed at every voxel returning the position  $x$  that attains the highest value. More details in Algorithm 3.

TTM-ref is simple but effective, in Sec. 4.1 it is shown that  $r_s = 3$  voxels is enough for correcting the shifts of the templates presented here. Contrarily to the TM algorithm, where computing the LNCC requires to sample the whole space of rotations and applied to all image positions. Here, the LNCC is only computed once per voxel at the optimal rotation predicted by TTM, and just at the neighborhood of the pre-detected positions. For the sake of efficiency, LNCC is now computed in the real space for every position by extracting a centered local patch of the image with the size of the template.

### 3.4 Implementation details

Integrating properly the needle over all rotations is a critical part of the procedure. In this work, this is accomplished by summing over a large sampling set of rotations. In the traditional approach, the exact distribution of rotations is not terribly critical, as we only retain the maximum response. However, when using the samples for integration, it is imperative that they are as uniformly distributed as possible, therefore we use the algorithm presented in [2].

TTM has been implemented for processing 3D images, tomograms, following the scheme depicted in Fig. 2. The blue boxes represent algorithms. *Tensorial Template* is Algorithm 1 and *Tensorial Field* is Algorithm 2. *Scalar Map* corresponds to compute  $\hat{c}_n(x)$ . *Find Peaks* is implemented by taking the  $P$  highest values of  $\hat{c}_n(x)$ , but considering a minimum exclusion distance among peaks in

---

**Algorithm 3** Position refinement for TTM (TTM-ref)

---

**Require:**  $f$ : image,  $t$ : template,  $C_n$ : tensorial field,  $L = \{x^i\}_{i=\{1,\dots,P\}}$ : list of  $P$  peaks obtained from  $\hat{c}_n$ ,  $r_s$ : sphere radius in voxels.

**Ensure:**  $L_{ref} = \{(\tilde{x}^i, R_{opt}^i)\}_{i=\{1,\dots,P\}}$ : list of  $P$  peaks with refined positions

```

for  $i \in P$  do                                     ▷ Loop for peaks
   $V_{x^i} \leftarrow$  all voxels in  $\{\|x^i - x\| \leq r_s\} \setminus x^i$ 
   $R_{opt} \leftarrow$  optimal rotation from  $C_n(x^i)$ 
   $c \leftarrow w(x^i)(f \star t'_R)(x^i)$                  ▷ LNCC
  for  $j \in V_{x^i}$  do
     $R \leftarrow$  optimal rotation from  $C_n(x^j)$ 
     $\tilde{c} \leftarrow w(x^j)(f \star t'_R)(x^j)$ 
    if  $\tilde{c} > c$  then
       $c \leftarrow \tilde{c}$ 
       $\tilde{x} \leftarrow x^j$ 
       $R_{opt} \leftarrow R$ 
    end if
  end for
   $L_{ref}^i \leftarrow (\tilde{x}, R_{opt})$ 
end for

```

---

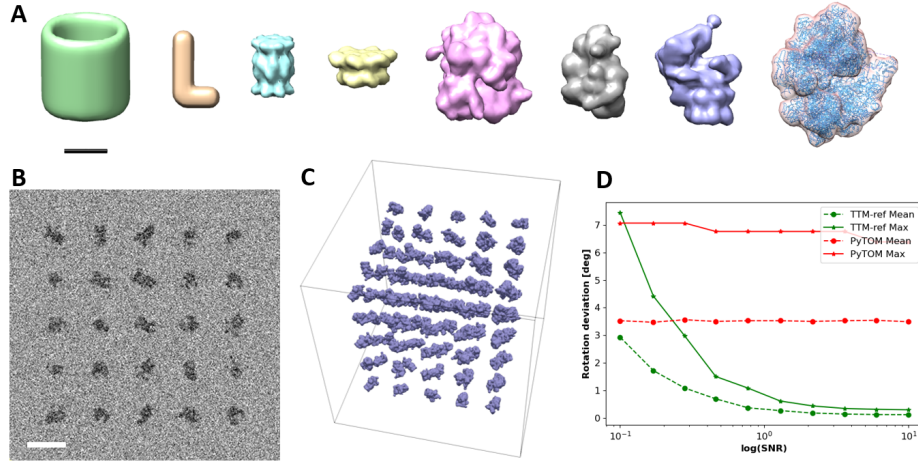
accordance with the template diameter, two times the mask radius. The implementation of *Find Peaks* also includes the heuristic for position refinement described in Sec. 3.3. A block processing scheme is proposed to avoid having many full copies with the dimensions of the original tomogram loaded in the main memory. For example, in cryo-ET the typical dimensions of a tomogram are around 1000x1000x250 voxels. Overlapping halos with template radius length is needed to prevent unreliable correlations on block borders. *Merge Peaks* gathers all peaks found in all blocks and sorts them to filter the  $P$  highest peaks in the whole image. The final output is a list of  $P$  template instances ranked by their similarity in terms of the LNCC with respect to the input template and including their pose, i.e., their rotation with respect to the input template. Finally, the user determines which peaks are real peaks just by defining a minimum threshold for the similarity.

## 4 Experimental results

All experiments are performed on an Ubuntu 20.04 workstation with a CPU Intel Xeon W2255 10-cores and 32GB RAM. In addition, TM implementation was accelerated with an Nvidia RTX4090 GPU.

### 4.1 TTM determines accurately positions and rotations

To validate the algorithms and quantitatively analyze their accuracy, we have prepared some synthetic tomograms. We have created 8 tomograms, one per each



**Fig. 3:** Synthetic data. (A) Isosurfaces of the 3D templates with 10 Å voxel size, from left to right: cylinder, L-shape, 3J9I, 3CF3, 4V4R, 1QvR, 4CR2 and 5MRC. The cylinder shows radial symmetry along the axis, 3J9I D7 and 3CF3 C6. 5MRC isosurface has transparency and also shows the atomic model used to generate the template density. Scale bar 100 Å. (B) 2D slice on XY-plane of the tomogram with SNR=0.1 containing instances of 4CR2 at random rotations. Scale bar 500 Å. (C) Isosurface of a noise free tomogram with 4CR2 instances. (D) Deviation in the rotations for the 4CR2 template at different *SNR* levels. Horizontal axis is in log scale.

template used in this study. Two templates are simple shapes, a cylinder and a L-shape, and the other six are taken from atomic models of the Protein Data Bank generating 3D densities with 10 Å voxel size (see Fig. 3.A). The proteins have been selected to represent different sizes and shapes, within the Protein Data Bank each entry is identified by a four digits code. Additionally, some templates present different symmetries. Each tomogram contains a 5x5x5 grid with 125 randomly rotated copies of the same template. The shape of all templates is cubic and has an odd size in order to generate templates with unambiguous centers. The size is derived from the bounding box of the structure, such that the center of the template corresponds to the center of mass of the structure, and the size of the template comprises the diagonal of the structure's bounding box ensuring that any rotation fits into it. An example of a tomogram is showed in Fig. 3.B-C.

The tomograms were processed with the two implementations proposed here, TTM and TTM-ref. For comparison, the tomograms were also processed with the widely used software in cryo-ET for TM, PyTOM (v1.1, Aug 29, 2023) [8,16,20].

Position accuracy is computed as the Euclidean distance in voxels between actual particle positions and algorithms output (see Tab. 1). The distance between two rotations expressed as quaternions,  $q_1$  and  $q_2$ , is measured by the angle  $2 \arccos(|q_1 \cdot q_2|)$  in degrees [24] (see Tab. 2).

The results in Tab. 1 show that, in general, the Frobenious norm is a good proxy for position detection since for most of the templates it achieves subvoxel accuracy, and in the rest, its shifting is always under 3 voxels. Regarding rotations, when positions are perfectly detected, TTM obtains an accuracy below  $0.3^\circ$ . Conversely, rotation accuracy of TM is limited by the sampling of  $SO(3)$ , in the case of PyTOM using 45123 samples, the results for maximum distances ranged  $4^\circ$  to  $13^\circ$ .

We observed that errors detecting positions greatly spoils rotation determination by TTM as advanced in Sec. 3.2. However, TTM-ref, TTM with position refinement (see Sec. 3.3), is able to compensate errors in positions, enabling also a highly accurate rotation determination for all templates. According with the results obtained, TTM accuracy is not affected by symmetry.

**Table 1:** Positions precision. Values are mean / maximum Euclidean distance in voxels between actual and predicted positions.

	<b>Cylinder</b>	<b>L-shape</b>	<b>3J9I</b>	<b>3CF3</b>
<b>PyTOM (TM)</b>	0.32 / 1.73	2.16 / 3.74	2.35 / 3.46	2.23 / 3.46
<b>TTM</b>	2.38 / 2.83	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0
<b>TTM-ref</b>	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0
	<b>4V4R</b>	<b>1QVR</b>	<b>4CR2</b>	<b>5MRC</b>
<b>PyTOM (TM)</b>	2.19 / 3.74	2.13 / 3.74	2.15 / 3.74	2.19 / 3.74
<b>TTM</b>	0.0 / 0.0	0.0 / 0.0	0.21 / 1.0	0.0 / 0.0
<b>TTM-ref</b>	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0

**Table 2:** Rotations precision. Values are mean / maximum angular distance in degrees between actual and predicted rotations.

	<b>Cylinder</b>	<b>L-shape</b>	<b>3J9I</b>	<b>3CF3</b>
<b>PyTOM (TM)</b>	2.20 / 4.09	5.70 / 12.56	2.48 / 4.69	2.51 / 5.14
<b>TTM</b>	0.03 / 0.06	31.40 / 39.01	0.03 / 0.12	0.06 / 0.17
<b>TTM-ref</b>	0.03 / 0.06	0.03 / 0.24	0.03 / 0.12	0.06 / 0.17
	<b>4V4R</b>	<b>1QVR</b>	<b>4CR2</b>	<b>5MRC</b>
<b>PyTOM (TM)</b>	3.22 / 5.27	3.84 / 7.07	3.49 / 6.39	3.02 / 5.20
<b>TTM</b>	0.14 / 0.26	0.09 / 0.24	0.97 / 7.60	0.07 / 0.16
<b>TTM-ref</b>	0.14 / 0.26	0.09 / 0.24	0.11 / 0.26	0.07 / 0.16

In Fig. 3.D, we add additive Gaussian noise to a synthetic tomogram (protein 4CR2) to analyze its impact on TTM and PyTOM performance. Deviations in rotations are under  $1^\circ$  for  $SNR > 1$  and reach a maximum around  $7^\circ$  in the

worst case for  $SNR = 0.1$ , mean deviations are under  $3^\circ$ , and under PyTOM values, up to  $SNR = 0.1$ . However, TTM seems to be more sensitive to noise than PyTOM. Positions are not displayed because they are always perfectly detected after refinement.

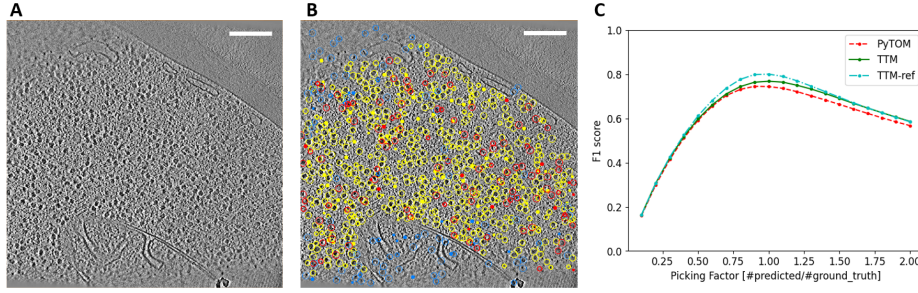
## 4.2 TTM outperforms TM for ribosome localization in cryo-ET

To evaluate the accuracy of TTM for object detection on real 3D images, specifically for ribosome detection on a set of cryo-tomograms, we have taken the dataset publicly available in EMPIAR-10988 [37]. This dataset contains 10 tomograms of *Schizosaccharomyces pombe* specimens at 13.68 Å. We took this dataset because it also includes a list with ribosome positions laboriously generated by experts, contrarily to other public datasets in cryo-ET. Currently, there is no method available to precisely identify the entire population of a macromolecule. TM is the only alternative to provide an estimation. However, these data have undergone careful processing and examination by experts in the field. The list of ribosome positions provided in this dataset was obtained through an initial particle picking using TM (PyTOM) with an intentional over-picking followed by manual refinement. In a meticulous and labor-intensive process, experts thoroughly review each individual instance to eliminate false positives and carefully examine the tomograms to avoid missing instances. Moreover, the authors were able to obtain several high-resolution averages from the instances finally isolated. Consequently, we can be confident that positions provided in this dataset actually correspond to real ribosome instances and represent quite well the whole population, so we can use them as ground truth.

Fig. 4 shows TTM performance for ribosome detection in EMPIAR-10988 dataset with respect to the ground truth and in comparison with TM. We use PyTOM as TM matching implementation, because it is nowadays the most popular in cryo-ET and has been largely improved during years, since its initial presentation [20] to its most recent version optimized for GPU hardware [8]. Here, we have used the ribosome density EMD-14411, obtained from EMPIAR-10988 dataset by subtomogram averaging, as template. An example of the cross-correlation map provided by TM,  $c_{R_{opt}}$ , and the Frobenious norm given by TTM,  $\hat{c}_n$ , is available in Supp. Mat. (see Fig. 1). To evaluate the detection performance, we compute the F1-score (harmonic mean of the precision and recall), assuming that a correct detection is a predicted peak closer than 10 voxels (approximately the ribosome radius) to a position in the ground truth list. TTM F1-score approximates 0.8 when the number of peaks nearly corresponds with the number of ground truth instances. That is, when picking factor, ratio between the number of predicted peaks and the ground truth, is 1. TTM-ref is a bit better but not so different. Notice that TTM exceeds PyTOM for the whole curve. Graphs for precision and recall are available in the Supp. Mat. (see Fig. 2).

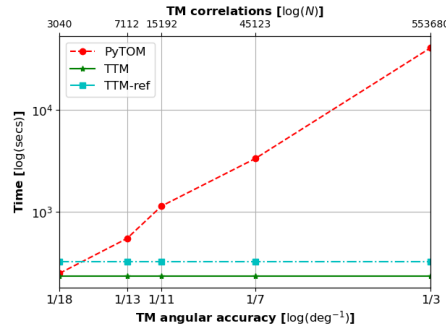
Recently, DL methods have been developed for macromolecular detection in cryo-ET, but they are limited to scenarios where reliable annotations are available for training and, unlike TM and TTM, none is able to determine the rotations for every detected instance. Conversely, TTM has been designed to be





**Fig. 4:** Object detection in a real dataset. (A) A 2D slice on XY-plane of a real tomogram containing a region of a *S. pombe* cell. (B) Overlaid on the 2D slice the true positives (yellow), false positives (red) and false negative (blue) detected by TTM. (C) TTM and PyTOM global F1-scores for the dataset EMPIAR-10988. (A-B) Scale bars 2000 Å.

an alternative to TM allowing to process datasets when only an input template is available. Therefore, it is not suitable to compare TTM with DL methods trained from exactly the same real dataset used for prediction. Instead, we take DeepFinder [31], the best performer in terms of macromolecular localization along the different editions of SHREC (Classification in Cryo-electron Tomograms) track [16,17]. Then, we train it with 10 fabricated tomograms including randomized (translations and rotations) copies of the ribosome template within realistic cellular contexts [30]. In addition, we also adapted DeepFinder to provide an output like TM and TTM cross-correlation enabling peaks extraction for different thresholds. Results are pretty similar to those obtained by TM and TTM (see Supp. Mat. Tab. 1) but DeepFinder (or any other DL approach) cannot assign rotations.



**Fig. 5:** Running times. Comparison between our implementation of TTM and a well-known GPU implementation of TM (PyTOM) for processing a single EMPIAR-10988 tomogram. Axes are in log scale.

### 4.3 TTM is faster than TM GPU implementations

Fig. 5 is the experimental counterpart of the complexity analysis presented in the introduction section of this paper (see Fig. 1). Fig. 5 shows the running times in log scales for processing a single tomogram of the real dataset used in Sec. 4.2. The effective dimensions, excluding padding voxels added during 3D reconstruction, of this tomogram are 960x928x230 voxels. Here, we compare the running times of the TTM implementations (with and without position refinement) against PyTOM, which provides different samplings of the rotations space obtaining different angular precision ranging from  $3^\circ$  to  $18^\circ$ . Despite PyTOM has recently been optimized to run on GPU architectures [8], it still requires processing times of several hours to achieve an angular precision below  $10^\circ$  because of the cubic computational complexity dependency with angular accuracy,  $O((360/\epsilon)^3)$ . Conversely, and without GPU acceleration, TTM takes less than four minutes to process a tomogram, 90 seconds more if the refinement is activated. TTM running time is independent of the angular accuracy, as its computational complexity is  $O(1)$ . TTM only uses the CPU, where the 35 correlations are distributed across various CPU cores as well as the computation of the eigenvectors on the detected peaks.

Generating the tensorial template is computationally costly (see Algorithm 1), it took 16 minutes for the ribosome template. But this is not relevant for processing time because tensorial template generation is independent of the tomograms to process.

## 5 Discussion

In this paper, we propose and implement a new computer method, which comprises several algorithms for the fast computation of the normalized correlation considering at the same time translation and rotation of the template. The method is based on the main theorem proposed and demonstrated in [3] but adapted and extended here for processing (tomograms). Our experimental results validate the novel mathematical approach based on Cartesian tensors.

We experimentally demonstrate that TTM is more accurate than TM for predicting positions and rotations in tomograms free of artifacts. Moreover, because of the complexity reduction of TTM with respect to TM, processing times for TTM are significantly lower than TM without the necessity of relying on GPU hardware for acceleration. Nowadays, the application of TM for tomography has basically been limited to cryo-ET, so we studied the performance of TTM in this domain. Now, TTM solves efficiently the problem of arbitrary object detection, besides rotation assignment, from just an input template, therefore TTM can also be used in application domains where TM computational cost was the limiting factor.

Furthermore, we demonstrate that a degree-4 tensor was sufficient to recover rotation information accurately in practice. As a future work, we plan to study in detail the impact of the tensor degree in the robustness against image distortions.

## Acknowledgments

A.M-S. was supported by the Ramon y Cajal Program of the Spanish State Research Agency (AEI) through the MICIU/AEI/10.13039/501100011033 and the European Union NextGenerationEU/PRTR under Grant RYC2021-032626-I and by the University of Murcia through the Attract-RYC 2023 Program.

## References

1. Al-Azzawi, A., Ouadou, A., Tanner, J.J., Cheng, J.: Autocryopicker: an unsupervised learning approach for fully automated single particle picking in cryo-em images. *BMC bioinformatics* **20**(1), 1–26 (2019)
2. Alexa, M.: Super-fibonacci spirals: Fast, low-discrepancy sampling of  $so(3)$ . In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 8291–8300 (2022)
3. Almira, J.M., Phelippeau, H., Martinez-Sanchez, A.: Fast normalized cross-correlation for template matching with rotations. *Journal of Applied Mathematics and Computing* (2024), <https://doi.org/10.1007/s12190-024-02157-6>
4. Applequist, J.: Traceless cartesian tensor forms for spherical harmonic functions: new theorems and applications to electrostatics of dielectric media. *Journal of Physics A: Mathematical and General* **22**(20), 4303 (1989)
5. Bepler, T., Morin, A., Rapp, M., Brasch, J., Shapiro, L., Noble, A.J., Berger, B.: Positive-unlabeled convolutional neural networks for particle picking in cryo-electron micrographs. *Nature methods* **16**(11), 1153–1160 (2019)
6. Böhm, J., Frangakis, A.S., Hegerl, R., Nickell, S., Typke, D., Baumeister, W.: Toward detecting and identifying macromolecules in a cellular context: template matching applied to electron tomograms. *Proceedings of the National Academy of Sciences* **97**(26), 14245–14250 (2000)
7. Cao, S., He, S., Li, Z., Wang, Z.: Extreme ratio between spectral and frobenius norms of nonnegative tensors. *SIAM Journal on Matrix Analysis and Applications* **44**(2), 919–944 (2023). <https://doi.org/10.1137/22M1502951>
8. Chaillet, M.L., van der Schot, G., Gubins, I., Roet, S., Veltkamp, R.C., Förster, F.: Extensive angular sampling enables the sensitive localization of macromolecules in electron tomograms. *International Journal of Molecular Sciences* **24**(17), 13375 (2023)
9. DiMaio, F.P., Soni, A.B., Phillips, G.N., Shavlik, J.W.: Spherical-harmonic decomposition for molecular recognition in electron-density maps. *International journal of data mining and bioinformatics* **3**(2), 205–227 (2009)
10. Dokmanic, I., Petrinovic, D.: Convolution on the  $n$ -sphere with application to pdf modeling. *IEEE transactions on signal processing* **58**(3), 1157–1170 (2009)
11. Fageot, J., Uhlmann, V., Püspöki, Z., Beck, B., Unser, M., Depeursinge, A.: Principled design and implementation of steerable detectors. *IEEE Transactions on Image Processing* **30**, 4465–4478 (2021)
12. Fäklér, F., Dimchev, G., Hodirnau, V.V., Wan, W., Schur, F.K.: Cryo-electron tomography structure of arp2/3 complex in cells reveals new insights into the branch junction. *Nature communications* **11**(1), 6437 (2020)
13. Freeman, W.T., Adelson, E.H., et al.: The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence* **13**(9), 891–906 (1991)

14. Gao, Z., Yi, R., Qin, Z., Ye, Y., Zhu, C., Xu, K.: Learning accurate template matching with differentiable coarse-to-fine correspondence refinement. *Computational Visual Media* **10**(2), 309–330 (2024)
15. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231* (2018)
16. Gubins, I., Chaillet, M.L., Schot, G.v.d., Trueba, M.C., Veltkamp, R.C., Förster, F., Wang, X., Kihara, D., Moebel, E., Nguyen, N.P., White, T., Bunyak, F., Papoulias, G., Gerolymatos, S., Zacharaki, E.I., Moustakas, K., Zeng, X., Liu, S., Xu, M., Wang, Y., Chen, C., Cui, X., Zhang, F.: SHREC 2021: Classification in Cryo-electron Tomograms. In: Biasotti, S., Dyke, R.M., Lai, Y., Rosin, P.L., Veltkamp, R.C. (eds.) *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association (2021). <https://doi.org/10.2312/3dor.20211307>
17. Gubins, I., Schot, G.v.d., Veltkamp, R.C., Förster, F., Du, X., Zeng, X., Zhu, Z., Chang, L., Xu, M., Moebel, E., Martinez-Sanchez, A., Kervrann, C., Lai, T.M., Han, X., Terashi, G., Kihara, D., Himes, B.A., Wan, X., Zhang, J., Gao, S., Hao, Y., Lv, Z., Wan, X., Yang, Z., Ding, Z., Cui, X., Zhang, F.: Classification in Cryo-Electron Tomograms. In: Biasotti, S., Lavoué, G., Veltkamp, R. (eds.) *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association (2019). <https://doi.org/10.2312/3dor.20191061>
18. Hillar, C.J., Lim, L.: Most tensor problems are np-hard. *J. ACM* **60**(6), 45:1–45:39 (2013). <https://doi.org/10.1145/2512329>, <https://doi.org/10.1145/2512329>
19. Hoffmann, P.C., Kreysing, J.P., Khusainov, I., Tuijtel, M.W., Welsch, S., Beck, M.: Structures of the eukaryotic ribosome and its translational states in situ. *Nature Communications* **13**(1), 7435 (2022)
20. Hrabe, T., Chen, Y., Pfeffer, S., Kuhn Cuellar, L., Mangold, A.V., Förster, F.: Pytom: A python-based toolbox for localization of macromolecules in cryo-electron tomograms and subtomogram analysis. *Journal of Structural Biology* **178**(2), 177–188 (2012). <https://doi.org/https://doi.org/10.1016/j.jsb.2011.12.003>, <https://www.sciencedirect.com/science/article/pii/S1047847711003492>, special Issue: Electron Tomography
21. Hu, H., Jiang, B., Li, Z.: Complexity and computation for the spectral norm and nuclear norm of order three tensors with one fixed dimension. *Arxiv* (2022), <https://arxiv.org/abs/2212.14775>
22. Huang, Q., Zhou, Y., Liu, H.F., Bartesaghi, A.: Accurate detection of proteins in cryo-electron tomograms from sparse labels. In: *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXI*. p. 644–660. Springer-Verlag, Berlin, Heidelberg (2022). [https://doi.org/10.1007/978-3-031-19803-8\\_38](https://doi.org/10.1007/978-3-031-19803-8_38), [https://doi.org/10.1007/978-3-031-19803-8\\_38](https://doi.org/10.1007/978-3-031-19803-8_38)
23. Huang, Q., Zhou, Y., Liu, H.F., Bartesaghi, A.: Weakly supervised learning for joint image denoising and protein localization in cryo-electron microscopy. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. pp. 3246–3255 (2022)
24. Huynh, D.Q.: Metrics for 3d rotations: Comparison and analysis. *J Math Imaging Vis* **35**, 155–164 (2009), <https://doi.org/10.1007/s10851-009-0161-2>
25. Jacob, M., Unser, M.: Design of steerable filters for feature detection using canny-like criteria. *IEEE transactions on pattern analysis and machine intelligence* **26**(8), 1007–1019 (2004)
26. Kofidis, E., Regalia, P.A.: On the best rank-1 approximation of higher-order supersymmetric tensors. *SIAM J. Matrix Anal. Appl.* **23**, 863–884 (2001)

27. Kolda, T.G., Mayo, J.: Shifted power method for computing tensor eigenpairs. *SIAM J. Matrix Anal. Appl.* **32**, 1095–1124 (2010)
28. Kozhasov, K., Tonelli-Cueto, J.: Probabilistic bounds on best rank-one approximation ratio. *Arxiv* (2022)
29. Lewis, J.P.: Fast template matching. In: *Vision interface*. vol. 95, pp. 15–19. Quebec City, QC, Canada (1995)
30. Martinez-Sanchez, A., Jasnin, M., Phelippeau, H., Lamm, L.: Simulating the cellular context in synthetic datasets for cryo-electron tomography. *bioRxiv* (2023). <https://doi.org/10.1101/2023.05.26.542411>, <https://www.biorxiv.org/content/early/2023/05/26/2023.05.26.542411>
31. Moebel, E., Martinez-Sanchez, A., Lamm, L., Righetto, R.D., Wietrzynski, W., Albert, S., Lariviere, D., Fourmentin, E., Pfeffer, S., Ortiz, J., Baumeister, W., Peng, T., Engel, B.D., Kervrann, C.: Deep learning improves macromolecule identification in 3d cellular cryo-electron tomograms. *Nature Methods* **18**(11), 1386–1394 (2021), <https://doi.org/10.1038/s41592-021-01275-4>
32. Nguyen, N.P., Ersoy, I., Gotberg, J., Bunyak, F., White, T.A.: Drpnet: automated particle picking in cryo-electron micrographs using deep regression. *BMC bioinformatics* **22**, 1–28 (2021)
33. Ni, T., Frosio, T., Mendonça, L., Sheng, Y., Clare, D., Himes, B.A., Zhang, P.: High-resolution in situ structure determination by cryo-electron tomography and subtomogram averaging using emclarity. *Nature protocols* **17**(2), 421–444 (2022)
34. Regalia, P., Kofidis, E.: The higher-order power method revisited: convergence proofs and effective initialization. In: *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*. vol. 5, pp. 2709–2712 vol.5 (2000). <https://doi.org/10.1109/ICASSP.2000.861047>
35. Rocco, I., Arandjelovic, R., Sivic, J.: Convolutional neural network architecture for geometric matching. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 6148–6157 (2017)
36. Roseman, A.M.: Particle finding in electron micrographs using a fast local correlation algorithm. *Ultramicroscopy* **94**(3-4), 225–236 (2003)
37. de Teresa-Trueba, I., Goetz, S.K., Mattausch, A., Stojanovska, F., Zimmerli, C.E., Toro-Nahuelpan, M., Cheng, D.W., Tollervey, F., Pape, C., Beck, M., et al.: Convolutional networks for supervised mining of molecular patterns within cellular context. *Nature Methods* **20**(2), 284–294 (2023)
38. Turk, M., Baumeister, W.: The promise and the challenges of cryo-electron tomography. *FEBS Letters* **594**(20), 3243–3261 (2020). <https://doi.org/https://doi.org/10.1002/1873-3468.13948>, <https://febs.onlinelibrary.wiley.com/doi/abs/10.1002/1873-3468.13948>
39. Vock, R., Dieckmann, A., Ochmann, S., Klein, R.: Fast template matching and pose estimation in 3d point clouds. *Computers & Graphics* **79**, 36–45 (2019)
40. Wagner, T., Merino, F., Stabrin, M., Moriya, T., Antoni, C., Apelbaum, A., Hagel, P., Sitsel, O., Raisch, T., Prumbaum, D., et al.: Sphire-cryolo is a fast and accurate fully automated particle picker for cryo-em. *Communications biology* **2**(1), 218 (2019)