3-D shape estimation of DNA molecules from stereo cryo-electron micro-graphs using a projection-steerable snake

Mathews Jacob, Thierry Blu, Cedric Vaillant, John Maddocks and Michael Unser

Abstract

We introduce a 3-D parametric active contour algorithm for the shape estimation of DNA molecules from stereo cryo-electron micrographs. We estimate the shape by matching the projections of a 3-D global shape model with the micrographs; we choose the global model as a 3-D filament with a B-spline skeleton and a specified radial profile. The active contour algorithm iteratively updates the B-spline coefficients, which requires us to evaluate the projections and match them with the micrographs at every iteration.

Since the evaluation of the projections of the global model is computationally expensive, we propose a fast algorithm based on locally approximating it by elongated blob-like templates. We introduce the concept of projection-steerability and derive a projection-steerable elongated template. Since the 2-D projections of such a blob at any 3-D orientation can be expressed as a linear combination of a few basis functions, matching the projections of such a 3-D template involves evaluating a weighted sum of inner-products between the basis functions and the micrographs. The weights are simple functions of the 3-D orientation and the inner-products are evaluated efficiently by separable filtering.

We choose an internal energy term that penalizes the average curvature magnitude. Since the exact length of the DNA molecule is known a-priori, we introduce a constraint energy term that forces the curve to have this specified length. The sum of these energies along with the image energy derived from the matching process is minimized using conjugate-gradients algorithm. We validate the algorithm using real as well as simulated data, and show that it performs well.

Index Terms

active contour, cryo, microscopy, ridge, spline, steerable, separable filtering

I. Introduction

Cryo-electron microscopy is an approach used to image bio-molecules such as DNA filaments [1]–[3]. The molecules are suspended in a thin layer of liquid, which is then cooled to a very low temperature. Thanks to the rapid cooling (of the order of 10^6 K/s), the resulting specimen can be considered to be a snapshot of its thermal oscillations. As compared to other approaches such as classical electron microscopy and atomic force microscopy, where the molecules are adsorbed onto supporting films, this method does not cause shape deformation. In this paper, we address the 3-D reconstruction of the shape of a DNA molecule from its stereo-micrographs (a typical pair of such images is shown in Fig. 1). This data is useful in probing the physical properties of the filament (such as its shape, stiffness, modes of oscillations, shape variation due to protein-bindings etc.) that play important roles in various bio-molecular

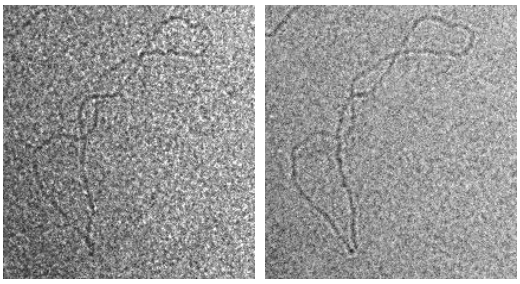


Fig. 1. Stereo views separated by 30° of a super-coiled DNA filament (1800 base pairs). Courtesy E. Larquet, Pasteur Institute, France.

processes.

Since exposure to electron beams causes degradation of the specimen, one usually restricts the number of views to two. Due to physical constraints, the angular separation between the views is limited to a maximum of 30 degrees. The micrographs also suffer from poor image contrast and low signal to noise ratio (SNR) due to the low electron dose. All these aspects make the reconstruction problem difficult.

The early approaches to this problem included manual reconstruction [4] and a semi-automatic search algorithm called the flying cylinder [5], [6]. In the manual scheme, the user clicks on the images to introduce pairs of corresponding points that define the curve; this is time consuming and not necessarily reproducible. The flying cylinder algorithm detects the filament by matching the projections (onto the image planes) of a 3-D cylindrical template with the stereo images. Since deriving the 2-D projections of a 3-D cylinder (with arbitrary orientation) was difficult, the authors approximated them with oriented rectangles. To reduce the number of matchings required, they discretized the orientation space and used a sequential search algorithm. The detected fragments were then sorted and interpolated to obtain a continuous curve. The performance of this algorithm is limited by the approximations, angular discretization, and, the multi-step strategy; in particular, the interpolation of the curve is only based on the detected fragments and is not necessarily consistent with the image data, nor the global optimum.

We address these shortcomings and propose a new algorithm that solves the 3-D reconstruction problem in a more exact and consistent manner using projection-steerable templates and a 3-D active contour model. An outline of the full procedure is given in Fig. 2. In an active contour framework, the estimation of the shape is formulated as an energy minimization problem. The snake energy is a linear combination of the image energy, the internal energy and the constraint energy terms (we discuss the details of the snake algorithm in Section IV). At each iteration of the optimization algorithm, the curve model is evaluated from its coefficients and the energy terms are computed based on the model and image information. The

curve coefficients are then updated so that the system converges towards the minimum of the energy function.

The image energy term, which is a crucial part of the snake energy, is a measure of the fit of the curve model with the image data. We consider a global model for the DNA filament whose skeleton is a 3-D parametric B-spline curve with a certain radial profile. Ideally, we would project the global model onto the projection planes and match the projections with the images (we compute the sum of the inner-products between the projections and the images) to obtain the fitness measure. Thanks to the linearity of the B-spline representation, the skeletons of the 2-D projections will be 2-D B-spline curves. However, its profile will be different at different 2-D curve points, depending of the orientation of the filament at the corresponding 3-D point. Thus the evaluation of the exact projections and performing the matching operation is computationally very expensive. Note that since we use an iterative optimization algorithm, the projections and matching procedure have to performed in a loop.

To reduce the computational complexity, we propose to approximate the global 3-D model locally as an elongated blob-like template. We introduce the concept of projection-steerablity, which is inspired by the work on 2-D orientation steerablity by Freeman et. al. [7]–[9]. We assume the radial profile[1] as a Gaussian and derive an elongated template in 3-D that is projection-steerable, i.e., the 2-D projections of this elongated template can be expressed as a linear combination of a few basis functions. With this framework, the matching of the projection of such a 3-D template can be performed inexpensively as a weighted sum of the inner-products between the basis functions and the images. The weights are simple functions of the orientation of the 3-D template and the inner-products are evaluated efficiently by separable filtering. We discuss the projection-steerable ridge detection in detail in Section III.

We show that the cubic B-spline representation is optimal for the description of smooth 3-D curve, provided it is parametrized with a constant arc-length. We also use the constant arc-length assumption to derive a simple expression for the internal energy. For this assumption to hold, we reparametrize the initial curve (derived from user inputs) such that the curve knots are uniformly spaced. Since the length of the DNA molecules are known a-priori, we use an additional constraint term that penalizes the curve for not having the specified length.

We use conjugate gradient algorithm for snake optimization. This scheme requires the efficient evaluation of the partial derivatives of the energy terms. Thanks to the projection-steerable templates and the curve representation using finitely supported B-spline functions, they are computed exactly and efficiently

[1]Here we approximate the DNA filament locally as an elongated Gaussian.

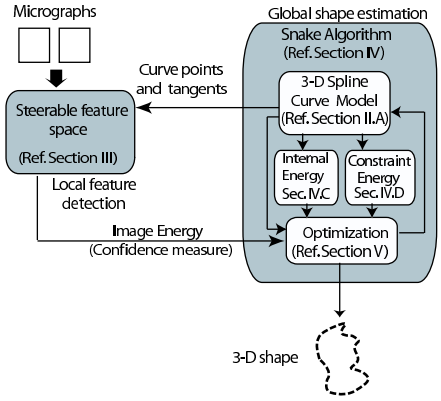


Fig. 2. Outline of the global 3-D shape estimation algorithm.

as shown in Section V.

II. Mathematical Preliminaries

A. Parametric representation of 3-D curve

A 3-D curve, (denoted as $\mathcal{C}$) can be described in terms of an arbitrary parameter t as $\mathbf{r}(t) = (x(t), y(t), z(t))$. When the curve is closed, the function vector, $\mathbf{r}(t)$, is periodic.

$\mathbf{r}(t)$ can be represented efficiently as a linear combination of some basis functions. Here, we focus on the B-spline curve representation [10]–[12] due to numerous advantages discussed in Section IV-C.1. Specifically, we represent the component functions of a 3-D curve in a uniform B-spline basis as

$$\mathbf{r}(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = \sum_{k=-\infty}^{\infty} \mathbf{c}(k)\, \beta^n(t-k), \tag{1}$$

where $\mathbf{c}(k) = [c_x(k), c_y(k), c_z(k)]$ is a sequence of coefficient vectors; in computer graphics, these are often called the control points [13]. The basis function β^n is the B-spline of degree n [14]. If the period, M, is an integer, we have $\mathbf{c}(k) = \mathbf{c}(k+M)$. This reduces the infinite summation to

$$\mathbf{r}(t) = \sum_{k=0}^{M-1} \mathbf{c}(k)\, \beta_p^n(t-k), \tag{2}$$

where β_p^n is the M - periodization of β^n:

$$\beta_p^n(t) = \sum_{k=-\infty}^{\infty} \beta^n(t - k\,M) \tag{3}$$

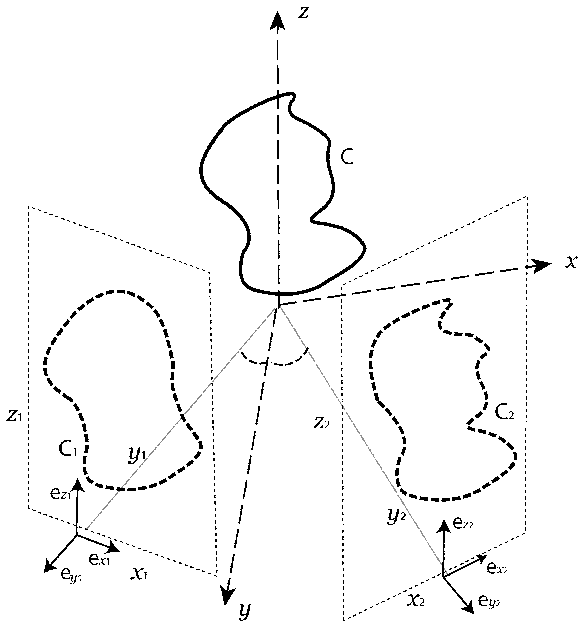


Fig. 3. 3-D curve and its 2-D projections.

Note that the special case of $n = 1$ (linear splines) yields a curve that is composed of line segments connecting the control points.

We denote the orthonormal basis vectors of the volume as $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$. The basis vectors of the i^{th} projection plane is $(\mathbf{e}_{x_i}, \mathbf{e}_{z_i})$, while the vector orthonormal to the plane is denoted by $\mathbf{e}_{y_i}$. An arbitrary vector $\mathbf{r}$, can be represented as

$$\mathbf{r} = x\,\mathbf{e}_x + y\,\mathbf{e}_y + z\,\mathbf{e}_z \tag{4}$$

$$= x_i\,\mathbf{e}_{x_i} + y_i\,\mathbf{e}_{y_i} + z_i\,\mathbf{e}_{z_i} \tag{5}$$

The projection of the vector $\mathbf{r}$ onto the plane is given by

$$\mathbf{r}_i = \mathbf{P}_i\,\mathbf{r} = x_i\,\mathbf{e}_{x_i} + z_i\,\mathbf{e}_{z_i}, \tag{6}$$

where $\mathbf{P}_i$ are the orthogonal projection matrices. The reconstruction algorithm requires projecting the curve model onto the image planes. Thanks to the linearity of the representation, the 2-D curve projections are also B-spline curves. The 2-D curve coefficients are:

$$\mathbf{c}_i(k) = \mathbf{P}_i\,\mathbf{c}(k), \tag{7}$$

Thus, the 2-D curve projections (we denote them by $\mathcal{C}_i$) are given by

$$\mathbf{r}_i(t) = \begin{bmatrix} x_i(t) \\ z_i(t) \end{bmatrix} = \sum_{k=0}^{M-1} \mathbf{c}_i(k)\, \beta_p^n(t-k), \tag{8}$$

where $\mathbf{r}_i(t) = \mathbf{P}_i \mathbf{r}(t)$. The projection matrix $\mathbf{P}_i$ can be thought of as the composition of a rotation matrix and a simple projection operator $\mathbf{P}$:

$$\mathbf{P}_i = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{P}} \mathbf{R}_i \tag{9}$$

Note that $\mathbf{P}$ is the same for all $\mathbf{P}_i$, while the projection geometry is specified by the rotation matrix $\mathbf{R}_i$, which performs the coordinate transformation from (x, y, z) to (x_i, y_i, z_i).

For the projection geometry shown in Fig. 3 ($\alpha_0 = -\alpha$, $\alpha_1 = \alpha$), the rotation matrices are

$$\mathbf{R}_1 = \mathbf{R}_z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

and $\mathbf{R}_2 = \mathbf{R}_z(-\alpha)$.

B. Orthogonal volume projection

The image formation in cryo-electron microscopy can be modeled as a a parallel projection. The assumption of parallel projection is justified by the large depth of field of the electron microscope; it is a well accepted hypothesis that is used implicitly in all 3D reconstructions of macro-molecules since 1982 [15], [16].

Modeling the measurement process as a line integral, we now obtain the expressions for the projection images from the 3-D volume data. We denote the volume by $f(\mathbf{r})$, where $\mathbf{r} = (x, y, z)$. The projected images $f_i = \mathcal{P}_i f$ are represented in the 2-D coordinate system where $\mathcal{P}_i$ denotes the orthogonal volume projection operator. Thus,

$$f_i(\mathbf{r}_i) = \int_{-\infty}^{\infty} f\left(\mathbf{P}_i^T \mathbf{r}_i + \gamma\, \mathbf{e}_{y_i}\right) d\gamma. \tag{11}$$

The above equation is easier to understand in the Fourier domain. The Fourier transform of f is given by

$$\hat{f}(\boldsymbol{\omega}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\mathbf{r})\, e^{j\langle \boldsymbol{\omega}, \mathbf{r}\rangle} dx dy dz, \tag{12}$$

where $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$. The Fourier transform of the function in the coordinate system (x_i, y_i, z_i) can be shown to be $\hat{f}\left(\mathbf{R}_i^T\, \boldsymbol{\omega}\right)$. The Fourier transform of the projection can be obtained by setting $\omega_{y_i} = 0$

(or by substituting $\boldsymbol{\omega} = \mathbf{P}^T \boldsymbol{\omega}_i$; $\boldsymbol{\omega}_i = (\omega_{x_i}, \omega_{z_i})$) in $\hat{f}\left(\mathbf{R}_i^T\, \boldsymbol{\omega}\right)$. Thus, the two-variable Fourier transform of the image is given by

$$\begin{aligned} \hat{f}_i\left(\boldsymbol{\omega}_i\right) &= \hat{f}\left(\mathbf{R}_i^T \mathbf{P}^T \boldsymbol{\omega}_i\right) \\ &= \hat{f}\left(\mathbf{P}_i^T \boldsymbol{\omega}_i\right) \end{aligned} \tag{13}$$

This expression can also be obtained using the Fourier-slice theorem [17].

III. Local filament detection

We have seen that the direct use of a global DNA model in the optimization algorithm can lead to a high computational complexity. Hence, we locally approximate the global model by an elongated blob. In this section, we address the detection of elongated blob-like structures in 3-D from their 2-D orthogonal projections (see Fig. 1). We addressed a similar problem in [18], where we derived the optimal rotation-steerable filters for 2-D feature detection. This method gave promising results for the detection of 2-D line-like structures. In this section, we generalize the concept of rotation-steerablity to projection-steerablity for 3-D filaments detection. This approach is well-suited for both the local scheme (where the detection is performed independently at each point) as well as the global approach (where the optimal orientation is specified by a model whose parameters are estimated).

A. *Projection-based feature detection*

Suppose our task is to check for the presence of an elongated 3-D blob—denoted by $f_{\mathrm{c}}\left(\mathbf{r}\right); \mathbf{r} \in \mathbb{R}^3$—with an unknown orientation, at a particular position $\mathbf{r}_{\mathrm{c}}$ in a 3-D volume f. The volume is known only through its orthogonal 2-D projections $f_i = \mathcal{P}_i f$. We formulate the detection procedure as a matched filtering; we consider a 3-D detector and match its orthogonal projections onto the image planes with the micrographs.

We choose the 3-D template[2] to be $h\left(\mathbf{r}\right) = f_{\mathrm{c}}\left(-\mathbf{r}\right)$ and denote its rotated versions by $h_{\mathbf{v}}\left(\mathbf{r}\right) = h\left(\mathbf{R}_{\mathbf{v}}\, \mathbf{r}\right)$, where $\mathbf{R}_{\mathbf{v}}$ is a 3-D rotation matrix. We use the sum of the inner-products between the 2-D template projections and the micrographs as the performance criterion:

$$C_{\mathbf{v}}\left(\mathbf{r}_{\mathbf{c}}\right) = \sum_{i=0}^{N-1} \left\langle f_i,\ \mathcal{P}_i\left(h_{\mathbf{v}}\right) \right\rangle \left(\mathbf{r}_{\mathbf{c},i}\right) \tag{14}$$

$$= \sum_{i=0}^{N-1} \left(f_i\left(\mathbf{r}_{\mathbf{c},i}\right) * \mathcal{P}_i\left(h_{\mathbf{v}}\left(-\mathbf{r}_{\mathbf{c},i}\right)\right)\right) \tag{15}$$

[2]We choose $h\left(\mathbf{r}\right) = f_{\mathrm{c}}\left(-\mathbf{r}\right)$ since we prefer to implement it using a convolution filtering operation.

where $\mathcal{P}_i;\quad i = 0 \ldots N-1$ are the orthogonal projection operators[3] and $\mathbf{r}_{\mathbf{c},i} = \mathbf{P}_i\,\mathbf{r}_{\mathbf{c}}$. Note that the criterion is a function of the orientation vector $\mathbf{v}$. If we perform the filament detection independently at each point, the optimal orientation vector $\mathbf{v}^*(\mathbf{r})$ and the likeliness measure $r^*(\mathbf{r})$ are given by

$$\mathbf{v}^*(\mathbf{r}) = \arg\max_{|\mathbf{v}|=1} \left(C_{\mathbf{v}}(\mathbf{r})\right) \tag{16}$$

$$r^*(\mathbf{r}) = C_{\mathbf{v}^*}(\mathbf{r}), \tag{17}$$

For an arbitrary 3-D template, the computation of the projections $\mathcal{P}_i(h_{\mathbf{v}})$ are expensive. To obtain the optimal orientation by numerical optimization, the template projections and their inner-products with the micrographs have to computed for each iteration; a direct implementation of the algorithm is not very practical, unless simplifying assumptions are made.

B. Projection-steerable ridge detection

To reduce the complexity in performing the projection matched filter detection, we use an approach similar to rotation-steerablity [7]–[9]. We would like to have a good 3-D filament detector whose projections (for any spatial orientation) are contained in a space spanned by a few basis functions. For such a detector, the evaluation of the performance criterion for each curve point simplifies to a weighted sum of the inner-products of the basis functions with the micrographs. The inner-products themselves can be efficiently pre-computed using 2-D filtering.

We now consider the family

$$V_{3\mathrm{D}} = \mathrm{span}\left\{\partial_{xx}\, g_3(\mathbf{r};\sigma), \partial_{yy}\, g_3(\mathbf{r};\sigma), \partial_{zz}\, g_3(\mathbf{r};\sigma), \partial_{xy}\, g_3(\mathbf{r};\sigma), \partial_{xz}\, g_3(\mathbf{r};\sigma), \partial_{yz}\, g_3(\mathbf{r};\sigma)\right\}, \tag{18}$$

where $g_D(\mathbf{r};\sigma) = \frac{1}{(2\pi\sigma)^{\frac{D}{2}}} \exp\left(-\frac{|\mathbf{r}|^2}{2\sigma^2}\right)$ is a D-dimensonal Gaussian and $\partial_{xy}\, f(\mathbf{r}) = \frac{\partial^2}{\partial x \partial y}(f(\mathbf{r}))$. We now show that any 3-D filter in this family is ideally suited for projection matched filter detection.

Proposition 1: The space $V_{3\mathrm{D}}$ is closed with respect to 3-D rotations.

[3]In our case $N = 2$, but the scheme is applicable for the general case as well.

Proof: The Fourier transforms of the basis functions are

$$\begin{aligned}
\partial_{xx}\, g_3(\mathbf{r};\sigma) &\overset{\mathcal{F}}{\leftrightarrow} -(2\pi)^{\frac{3}{2}}\, \omega_x^2\, g_3\left(\boldsymbol{\omega};\sigma^{-1}\right)\\
\partial_{yy}\, g_3(\mathbf{r};\sigma) &\overset{\mathcal{F}}{\leftrightarrow} -(2\pi)^{\frac{3}{2}}\, \omega_y^2\, g_3\left(\boldsymbol{\omega};\sigma^{-1}\right)\\
\partial_{zz}\, g_3(\mathbf{r};\sigma) &\overset{\mathcal{F}}{\leftrightarrow} -(2\pi)^{\frac{3}{2}}\, \omega_z^2\, g_3\left(\boldsymbol{\omega};\sigma^{-1}\right)\\
\partial_{xy}\, g_3(\mathbf{r};\sigma) &\overset{\mathcal{F}}{\leftrightarrow} -(2\pi)^{\frac{3}{2}}\, \omega_x\omega_y\, g_3\left(\boldsymbol{\omega};\sigma^{-1}\right)\\
\partial_{xz}\, g_3(\mathbf{r};\sigma) &\overset{\mathcal{F}}{\leftrightarrow} -(2\pi)^{\frac{3}{2}}\, \omega_x\omega_z\, g_3\left(\boldsymbol{\omega};\sigma^{-1}\right)\\
\partial_{yz}\, g_3(\mathbf{r};\sigma) &\overset{\mathcal{F}}{\leftrightarrow} -(2\pi)^{\frac{3}{2}}\, \omega_y\omega_z\, g_3\left(\boldsymbol{\omega};\sigma^{-1}\right)
\end{aligned}$$

where $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$. Since the basis functions are the products of second degree monomials with a Gaussian in the Fourier domain, an arbitrary function in $V_{3\mathrm{D}}$ is a second degree polynomial multiplied by a Gaussian (the six homogenous monomials is a basis of the space of homogeneous polynomials of degree 2). It can be written in a compact form as

$$\hat{h}(\boldsymbol{\omega}) = (2\pi)^{\frac{3}{2}}\, \left(\boldsymbol{\omega}^T \mathbf{A}\, \boldsymbol{\omega}\right) g_3\left(\boldsymbol{\omega};\sigma^{-1}\right) \tag{19}$$

Here, $\mathbf{A}$ is a symmetric 3x3 coefficient matrix that characterizes the shape of h[4]. The Fourier transform of a $\mathbf{R}$-rotated version[5] of h is given by

$$\hat{h}(\mathbf{R}\boldsymbol{\omega}) = (2\pi)^{\frac{3}{2}}\, \left(\boldsymbol{\omega}^T \underbrace{\mathbf{R}^T\mathbf{A}\mathbf{R}}_{\mathbf{A}_\mathbf{R}}\boldsymbol{\omega}\right) g_3\left(\boldsymbol{\omega};\sigma^{-1}\right), \tag{20}$$

where $\mathbf{R}$ is the 3x3 rotation matrix. Note that in the above step, we have used the isotropy of the function $\hat{g}$. Since the new filter $h(\mathbf{R}\mathbf{r})$ has the same form as (19) for any rotation matrix $\mathbf{R}$, it is still in $V_{3\mathrm{D}}$. ■

Proposition 2: The orthogonal projection $\mathcal{P}_i$ of the space $V_{3\mathrm{D}}$ onto a plane is the function space $V_{2\mathrm{D},i}$:

$$V_{2\mathrm{D},i} = \mathrm{span}\left\{\partial_{x_ix_i}\, g_2(\mathbf{r}_i;\sigma), \partial_{z_iz_i}\, g_2(\mathbf{r}_i;\sigma), \partial_{x_iz_i}\, g_2(\mathbf{r}_i;\sigma)\right\}. \tag{21}$$

Proof: The Fourier transform of the projection of an arbitrary function in $V_{3\mathrm{D}}$ is obtained by substituting (19) in (13):

$$\hat{h}_i(\boldsymbol{\omega}_i) = \sqrt{2\pi}\, \left(\boldsymbol{\omega}_i^T \underbrace{\mathbf{P}_i\mathbf{A}\mathbf{P}_i^T}_{\mathbf{B}_i}\boldsymbol{\omega}_i\right) \underbrace{2\pi\, g_3\left(\mathbf{P}_i^T\boldsymbol{\omega}_i;\sigma^{-1}\right)}_{g_2(\boldsymbol{\omega}_i;\sigma^{-1})} \tag{22}$$

Since the 2x2 matrix $\mathbf{B}_i$ is symmetric, $\hat{h}_i(\boldsymbol{\omega}_i)$ is a second degree polynomial in $\boldsymbol{\omega}_i$. This implies that h_i is a linear combination of the functions $\partial_{xx}\, g_2(\mathbf{r}_i,\sigma)$, $\partial_{xy}\, g_2(\mathbf{r}_i,\sigma)$ and $\partial_{yy}\, g_2(\mathbf{r}_i;\sigma)$. Thus $\mathcal{P}_i(V_{3\mathrm{D}}) \subseteq V_{2\mathrm{D,i}}$.

[4]We have made use of the fact the the Fourier transform of a Gaussian function is indeed a Gaussian.

[5]We use the property that the rotation of the filter is equivalent to rotating its Fourier transform.

We also have the relations $\mathcal{P}_i\left(\partial_{xx}\, g_3\right) = \partial_{x_i x_i}\, g_2$, $\mathcal{P}_i\left(\partial_{zz}\, g_3\right) = \partial_{z_i z_i}\, g_2$ and $\mathcal{P}_i\left(\partial_{xz}\, g_3\right) = \partial_{x_i z_i}\, g_2$. They imply that $V_{\mathrm{2D,i}} \subseteq \mathcal{P}_i\left(V_{\mathrm{3D}}\right)$. Thus we have $V_{\mathrm{2D,i}} = \mathcal{P}_i\left(V_{\mathrm{3D}}\right)$

■

We have seen that V_{3D} is closed under 3-D rotations. Hence, if we choose a 3-D detector in this space, its rotated versions are guaranteed to be in the same space. We have also seen that $\mathcal{P}_i\left(V_{\mathrm{3D}}\right) = V_{\mathrm{2D,i}}$, which implies that the projection of any rotated version of the detector is in $V_{\mathrm{2D,i}}$. Moreover, since the functions in $V_{\mathrm{2D,i}}$ are bandpass, the detection scheme will not be sensitive to smooth intensity variations that are common with micrographs[6].

In this paper, we have restricted ourselves to second order detectors for simplicity. However the concept of projection-steerablity is more general: any 3-D function that can be represented as a linear combination of the differentials (up to a certain order) of an isotropic function is projection-steerable.

C. 3-D ridge detection

We have seen that V_{3D} is ideally suited for projection-steerable matching. Hence, we would like to choose the most elongated blob-like structure in this space as our local 3-D template. We derive the optimally elongated local template in Appendix-A as

$$h\left(\mathbf{r}\right) = \sqrt{\frac{3}{20}}\left(\partial_{yy}\, g_3\left(\mathbf{r},\sigma\right) + \partial_{zz}\, g_3\left(\mathbf{r},\sigma\right) - \frac{2}{3}\,\partial_{xx}\, g_3\left(\mathbf{r},\sigma\right)\right) \tag{23}$$

See 3-D plots of this detector in Fig. 4.

Neglecting the normalization constant, we rewrite the expression for the optimal filter (23) oriented along the unit vector $\mathbf{v}$ as

$$h_{\mathbf{v}}\left(\mathbf{r}\right) = \underbrace{\left(\partial_{xx} + \partial_{yy} + \partial_{zz}\right)\, g_3\left(\mathbf{r};\sigma\right)}_{\text{Laplacian of } g\left(\mathbf{r};\sigma\right)} - \frac{5}{3}\,\partial_{\mathbf{vv}}\, g_3\left(\mathbf{r};\sigma\right), \tag{24}$$

where $\partial_{\mathbf{vv}} f\left(\mathbf{r}\right) = \frac{\partial^2}{\partial\gamma^2} f\left(\mathbf{r} + \gamma\,\mathbf{v}\right)$. Note that the Fourier transform of the filter is given by (19) with $\mathbf{A} = \mathbf{I}_3 - \frac{5}{3}\,\mathbf{v}\mathbf{v}^T$, where $\mathbf{I}_3$ is the 3x3 identity matrix. By substituting (24) in (22) and by performing the manipulations shown in Appendix-B, we get

$$\mathcal{P}_i\left(h_{\mathbf{v}}\left(\mathbf{r}\right)\right) = \mathbf{v}^T\left[\mathbf{R}_i^T\,\mathbf{G}_i\left(\mathbf{r}_i;\sigma\right)\mathbf{R}_i\right]\mathbf{v}, \tag{25}$$

[6]In traditional schemes, these variations are removed by a high-pass preprocessing filter [5].

where

$$\mathbf{G}_i(\mathbf{r}_i;\sigma) = \begin{bmatrix} \left(\partial_{z_iz_i} - \frac{2}{3}\partial_{x_ix_i}\right) g_2(\mathbf{r}_i;\sigma) & 0 & -\left(\frac{5}{3}\partial_{x_iz_i}\right) g_2(\mathbf{r}_i;\sigma) \\ 0 & \left(\partial_{x_ix_i} + \partial_{z_iz_i}\right) g_2(\mathbf{r}_i;\sigma) & 0 \\ -\left(\frac{5}{3}\partial_{x_iz_i}\right) g_2(\mathbf{r}_i;\sigma) & 0 & \left(\partial_{z_iz_i} - \frac{2}{3}\partial_{x_ix_i}\right) g_2(\mathbf{r}_i;\sigma) \end{bmatrix} \tag{26}$$

and in the application we consider here $\{\mathbf{R}_i\}_{i=1,2}$ are the rotation matrixes given by (10).

Note that, when $\mathbf{R}_i\mathbf{v} = (1,0,0)$ (horizontal filament parallel to the image plane), we get $\mathcal{P}_i(h_\mathbf{v}) = g_{z_iz_i}(\mathbf{r}_i;\sigma) - \frac{2}{3}g_{x_ix_i}(\mathbf{r}_i;\sigma)$—an elongated detector. On the other hand, if $\mathbf{R}_i\mathbf{v} = (0,1,0)$ (i.e., the filament is orthogonal to the image plane so that its projection is an isotropic blob rather than a filament), we get $\mathcal{P}_i(h_\mathbf{v}) = g_{x_ix_i}(\mathbf{r}_i;\sigma) + g_{z_iz_i}(\mathbf{r}_i;\sigma)$—the isotropic Laplacian detector. In other words, we use different 2-D detectors on the image planes, depending on the spatial orientation of the 3-D template. We give some examples with the 3-D template and their projections in Fig. 4.

Using (25), we simplify the expression (14) of $C_\mathbf{v}(\mathbf{r})$, which is to be minimized, to

$$C_\mathbf{v}(\mathbf{r}) = \mathbf{v}^T \underbrace{\left[\sum_{i=0}^{N-1} \mathbf{R}_i^T\, \mathbf{H}_{f_i}(\mathbf{r}_i)\, \mathbf{R}_i\right]}_{\mathbf{H}_{3D}(\mathbf{r})} \mathbf{v} \tag{27}$$

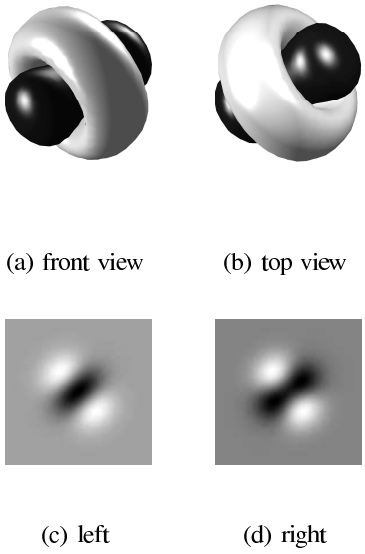
where

$$\mathbf{H}_{f_i}(\mathbf{r}_i) = f_i * \mathbf{G}_i(\mathbf{r}_i;\sigma)\,. \tag{28}$$

Here, we used the property $G_i(\mathbf{r}) = G_i(-\mathbf{r})$. Thus, we can match the image locally with the projection of a 3-D blob oriented along $\mathbf{v}$ by evaluating (27). To compute $\mathbf{H}_{f_i}(\mathbf{r}_i)$ at every point in the micrograph, we need to filter the image with the templates corresponding to the 5 non-zero entries of (26). Since these entries are linear combinations of the three functions $\partial_{x_ix_i} g_2(\mathbf{r}_i;\sigma), \partial_{z_iz_i} g_2(\mathbf{r}_i;\sigma)$ and $\partial_{x_iz_i} g_2(\mathbf{r}_i;\sigma)$, it is in fact sufficient to filter the images with three separable templates. The entries of $\mathbf{H}_{f_i}(\mathbf{r}_i)$ are linear combinations of these filtered signals.

IV. Constrained reconstruction of DNA filaments using the 3-D snake model

In the previous section, we have addressed the local detection of elongated 3-D blobs. In this section, we combine the local likeliness measures into the global model and estimate its parameters. Since the skeleton of the DNA molecule is represented by a curve, the well established framework of active contour models [19], [20] is most appropriate for this purpose.



(a) front view (b) top view

(c) left (d) right

Fig. 4. (a) and (b) Isosurface plots of the front (viewed from the x-z plane) and top (viewed from the x-y plane) views of a 3-D detector oriented at 30° to the x- axis and 30° to the x-y plane ($\theta = 30^\circ, \phi = 30^\circ$). (c) and (d) Projections of the 3-D filter onto the image planes oriented at -15° and -15° to the y axis

A. Active contour algorithm: Formulation

Traditionally, snakes or active contour models were introduced for the segmentation of closed objects in images. The popularity of these schemes may be attributed to their ability to aid the segmentation process with a-priori knowledge and user interaction. Snakes, as introduced in the seminal work of Kass et. al., are smooth curve models that evolve from an initial guess towards some boundary in the image such that some energy functional is minimized [19], [20].

These models were extended to 3-D for the estimation of coronary vessel centerlines from X-ray angiographic projections [21], [22]. This approach considers the evolution of a 3-D curve so that its 2-D projections onto the respective planes match the images. The matching is performed using distance maps or gradient vector flow fields and is done in the more general setting of perspective projection.

Here, we propose a refined approach, which mainly differs in the matching procedure and the curve representation used. Our criterion is obtained by projecting the optimal templates, oriented along the curve tangents, onto the image planes and matching them with the micrographs. Note that the shapes of the template projections depend on the 3-D curve tangent directions.

Before going into the details of the algorithm, we briefly review the fundamentals of the snake algorithm. Since the final shape is determined by the minimum of the snake energy, its choice deserves

Energy Type	Expression
Image Energy	$E_{\text{image}} = -\oint_{\mathcal{C}} d\mathbf{r}^T \, \mathbf{H}_{3\text{D}}(\mathbf{r}) \, d\mathbf{r}$ (ref Eq. (32))
Internal Energy	$E_{\text{int}} = \frac{1}{k^2} \int_0^M \left\|\mathbf{r}''(t)\right\|^2 dt$ (ref Eq. (34))
Length Constraint	$E_{\text{const}} \left(\int_0^M \left\|\mathbf{r}'(t)\right\| dt - \text{Length} \right)^2$ (ref Eq. (38))
Point Constraint	$E_{\text{const}} = \sum_{i=0}^{N_c-1} \min_{t\in[0,M]} \left\|\mathbf{r}(t) - \mathbf{r}_{c,i}\right\|^2$ (ref Eq. (39))

TABLE I

DIFFERENT ENERGY TERMS USED IN THE SNAKE OPTIMIZATION

proper attention. Similar to conventional snakes, we choose the energy functional as a linear combination of three separate terms.

1) The image energy, which is responsible for guiding the snake towards the filament.
2) The internal energy, which ensures that the extracted shape of the filament is smooth.
3) The constraint energy, which enables the user to enforce extra constraints such as the curve length.

The total energy of the snake is written as

$$E_{\text{snake}}(\mathbf{\Theta}) = E_{\text{image}}(\mathbf{\Theta}) + \mu_{\text{int}} \, E_{\text{int}}(\mathbf{\Theta}) + \mu_{\text{const}} \, E_{\text{const}}(\mathbf{\Theta}), \tag{29}$$

where $\mathbf{\Theta}$ is the collection of curve coefficients $\mathbf{\Theta} = \{\mathbf{c}(k);\ k = 0, \ldots M-1\}$. Here μ_{int} and μ_{const} are weights used to balance the different energy terms. The optimal curve parameters are obtained as

$$\overline{\mathbf{\Theta}} = \arg\min_{\mathbf{\Theta}} E_{\text{snake}}(\mathbf{\Theta}) \tag{30}$$

We describe the different energy terms in detail in the following subsections.

B. Image Energy

The image energy term is a measure of the fit of the model to the image data. Consider a point $\mathbf{r}(t)$ on the planar curve $\mathcal{C}$. The tangent vector of the curve at $\mathbf{r}(t)$, given by $d\mathbf{r}(t) = \left(\frac{dx(t)}{dt}, \frac{dy(t)}{dt}, \frac{dz(t)}{dt}\right) dt$,

defines the direction of the elongated blob at that point. We define the likeliness of a blob at the curve point $\mathbf{r}(t)$, oriented along $d\mathbf{r}(t)$, as

$$E_{\text{goodness}}(\mathbf{r}) = d\mathbf{r}(t)^T \ \mathbf{H}_{3\text{D}}(\mathbf{r}) \ d\mathbf{r}(t). \tag{31}$$

Recall from subsection III-B that this quantity is equivalent to projecting the optimal 3-D detector, oriented along $d\mathbf{r}$, onto the projection planes and then computing the sum of the square errors between the template projections and the micrographs. Note that if $d\mathbf{r} = \mathbf{v}^*$ as in (16), we get $E_{\text{goodness}}(\mathbf{r}) = r^*(\mathbf{r})$ which is the maximum possible value.

We obtain the likeliness of the entire curve by integrating the goodness measures along the curve.

$$E_{\text{image}}(\Theta) = -\int_0^M \left(d\mathbf{r}(t)^T \ \mathbf{H}_{3\text{D}}(\mathbf{r}) \ d\mathbf{r}(t) \right) dt, \tag{32}$$

The negative sign is introduced since the curve evolution is posed as an energy minimization problem. By using (30) to obtain the optimal coefficients, we are jointly estimating the optimal orientations and magnitudes at the voxels through which the curve passes. Note that the optimal orientation at each curve point is dependent on the optimal coefficients indirectly through the curve model (i.e. the tangent to the curve). Since the number of curve coefficients is typically much less than the number of voxels through which the curve passes, this scheme is more robust than a local approach. An advantage of the criterion (32) is that it is independent of the curve parameter t. Evolving the curve using such a measure will not cause the parametrization to change during the optimization process, thus preserving the curve stiffness[7].

C. Internal Energy

The internal energy term is responsible for ensuring the smoothness of the reconstructed shape. It is essentially a regularization term that penalizes non-smooth shapes, thus making the reconstruction problem better conditioned. The smoothness of the curve can be quantified by its total curvature magnitude; a stiff curve will have a low value of mean curvature magnitude. The curvature of the curve at a point $\mathbf{r}(t)$ is defined as

$$|\kappa(\mathbf{r})|^2 \ = \ \left| \frac{\mathbf{r}'(t) \times \mathbf{r}''(t)}{|\mathbf{r}'(t)|^3} \right|^2, \tag{33}$$

where $\mathbf{r}'(t) = (x'(t), y'(t), z'(t))$ is the derivative vector and $\mathbf{r}''(t) = (x''(t), y''(t), z''(t))$ is the vector of second differentials. Using the expression of the average curvature magnitude—$\int_0^M |\kappa(\mathbf{r})|^2 \, dt$—directly as the internal energy leads to complicated expressions for the partial derivatives. Using standard

[7]Many snake energies are parameter-dependent, causing the curve knots to accumulate at points of high edge strength.

results from differential geometry [23], we simplify this term to (see also [12])

$$\int_0^M |\kappa(\mathbf{r})|^2 \, dt \;=\; \frac{1}{k^2} \int_0^M \left|\mathbf{r}''(t)\right|^2 dt \tag{34}$$

provided

$$\left|\mathbf{r}'(t)\right|^2 = k, \quad \forall t; \tag{35}$$

that is, when the curve is parametrized by its curvilinear abscissa. Here

$$k = \frac{1}{M} \left(\underbrace{\int_0^M \left|\mathbf{r}'(t)\right| dt}_{\text{Length}} \right)^2 \tag{36}$$

is the total length per unit value of the parameter.

Since the uniform B-spline curve has its knots at the integer parameter values, (35) requires that the knots be uniformly spaced on the curve. Thus the smoothness term (34) is inversely proportional to the fourth power of the distance between the knots; the curve will be smooth if its knots are well separated. We will see in Section V-B that the partial derivatives of the (right hand side) r.h.s of (34) are much easier to compute than those of its (left hand side) l.h.s. To ensure that (35) hold, we resample the initial curve (obtained by user initialization) such that we have constant arc-length; the user input the initial curve by entering corresponding pairs[8] of points on the stereo views.

1) Choice of the basis function: Using the well-known variational properties of B-splines [24], we can show that the minimization of $\oint_C |\mathbf{r}''|^2$ subject to interpolation constraints gives a cubic spline curve. Thus, the cubic B-spline representation appears to be the natural choice, for it gives minimum curvature curves when the knots are uniformly spaced. The use of spline curves also brings in additional gains due to the existence of efficient algorithms [14], the local control of the contour due to the finite support of the B-spline basis function, and their good approximation properties [25].

2) Internal energy term: We reparametrize the initial curve (derived from the interpolation of the user input points) so that the knot points are uniformly spaced. Thanks to the parameter independent image energy term, we can safely assume that the curve will remain approximately in the constant arc-length parametrization. Hence, we choose the internal energy term as

$$E_{\text{int}} \;=\; \frac{1}{k^2} \int_0^M \left|\mathbf{r}''(t)\right|^2 dt. \tag{37}$$

Recall from (34) that using this term as the internal energy is equivalent to minimizing the average square magnitude of the curvature.

[8]the corresponding pairs are constrained to have the save z value.

D. External constraint energy.

As mentioned before, the external constraint energy is a means for the user to enforce extra constraints on the reconstruction. We use two constraint terms in our implementation.

1) Length constraint: The length of the DNA filaments are known a-priori; this information can be imposed on the reconstruction process to make it more robust. We introduce this constraint into the framework by penalizing the term

$$E_{\text{const}} = \left(\int_0^M \left| \mathbf{r}'(t) \right| dt - \text{Length} \right)^2, \tag{38}$$

where Length is the expected length of the molecule.

2) Point constraint: We use a point constraint to enable the user to aid the reconstruction process; he can specify a few 3-D points that should lie on the final shape. This constraint is basically the sum of the distances between these points and the closest points on the curve. The constraint energy is given by

$$E_{\text{const}} = \sum_{i=0}^{N_c - 1} \min_{t \in [0,M]} \left| \mathbf{r}(t) - \mathbf{r}_{c,i} \right|^2, \tag{39}$$

where $\mathbf{r}_{c,i}; i = 0, \ldots, N_c - 1$ are the constraints. This approach can be thought of as introducing virtual springs that pull the curve towards the desired points. One end of the spring is fixed to the constraint point, while the other end slides on the curve.

V. Curve evolution: the optimization algorithm

As mentioned before, the snake algorithm evolves the curve from its initial position to the final shape using energy minimization. Since the individual energy terms are non-linear functions of the curve coefficients, we require a numerical optimization algorithm. We use the conjugate gradient algorithm to refine the initial guess derived from the user inputs. The user specifies pairs of corresponding points on the stereo images that are then interpolated to derive the initial 3-D curve. This curve is later resampled to a specified number of knot points (since the approximation ability is decided by the number of knot points) such that (35) is satisfied. A summary of the whole algorithm is given in Fig 5.

The optimization scheme requires the evaluation of the partial derivatives of the snake energy. Since these quantities have to be repeatedly evaluated in the iteration loop, their computational complexity will determine the time taken by the snake algorithm. In this section, we derive efficient expressions for the derivatives of the individual energy terms.

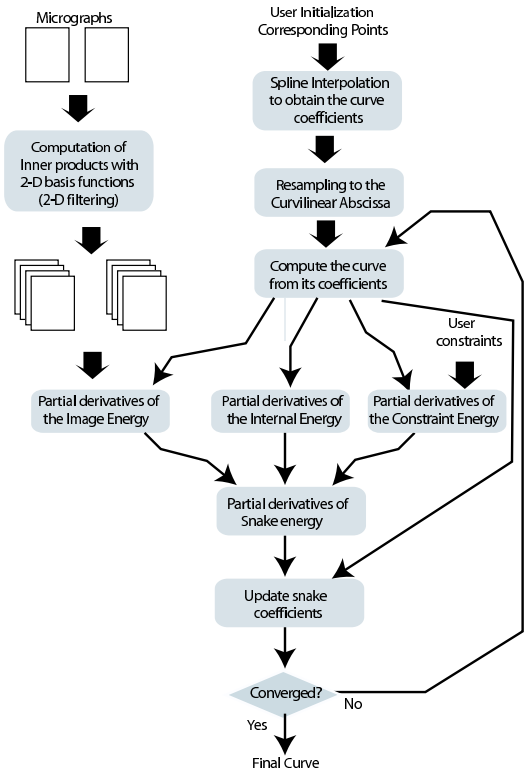


Fig. 5. Block diagram of the snake optimization algorithm.

A. *Partial derivatives of the image energy*

Differentiating (32) with respect to the coefficient $c_x(k)$ and applying the chain rule yields

$$\frac{\partial}{\partial c_x(k)} E_{\text{image}} = \underbrace{-2\int_0^M \left(d\mathbf{r}(t)^T \mathbf{H}_{3\text{D}}(\mathbf{r}(t)) \begin{bmatrix} \beta_p^{n\prime}(t-k) \\ 0 \\ 0 \end{bmatrix} \right) dt}_{I_1} - \underbrace{\int_0^M \left(d\mathbf{r}^T(t) \mathbf{H}_{3\text{D,x}}(\mathbf{r}(t)) d\mathbf{r}(t) \beta_p^n(t-k) \right) dt}_{I_2}, \quad (40)$$

where $\mathbf{H}_{3\text{D,x}}$ is a 3x3 matrix whose entries are the partial derivatives of the corresponding entries of $\mathbf{H}_{3\text{D}}$ with respect to x. We now focus on obtaining the expression of $\mathbf{H}_{3\text{D,x}}$

$$\begin{aligned}
\mathbf{H}_{\mathrm{3D,x}}(\mathbf{r}) &= \sum_{i=0}^{N-1} \mathbf{R}_i^T \left(\frac{\partial}{\partial x} \mathbf{H}_{f_i}(\mathbf{r}_i) \right) \mathbf{R}_i \\
&= \sum_{i=0}^{N-1} \mathbf{R}_i^T \left(\underbrace{\frac{\partial}{\partial x_i} \mathbf{H}_{f_i}}_{\mathbf{H}_{f_i,x_i}}(\mathbf{r}_i) \ \frac{\partial x_i}{\partial x} + \underbrace{\frac{\partial}{\partial z_i} \mathbf{H}_{f_i}}_{\mathbf{H}_{f_i,z_i}}(\mathbf{r}_i) \ \frac{\partial z_i}{\partial x} \right) \mathbf{R}_i \qquad (41) \\
&= \sum_{i=0}^{N-1} \left(\mathbf{R}_i^T \mathbf{H}_{f_i,x_i}(\mathbf{r}_i) \mathbf{R}_i \right) \underbrace{\frac{\partial x_i}{\partial x}}_{\mathbf{P}_i(0,0)} + \left(\mathbf{R}_i^T \mathbf{H}_{f_i,z_i}(\mathbf{r}_i) \mathbf{R}_i \right) \underbrace{\frac{\partial z_i}{\partial x}}_{\mathbf{P}_i(1,0)} \qquad (42)
\end{aligned}$$

Here, the matrices $\mathbf{H}_{f_i,x_i}$ and $\mathbf{H}_{f_i,z_i}$ are

$$\begin{aligned}
\mathbf{H}_{f_i,x_i}(\mathbf{r}_i) &= f * \begin{bmatrix} \left(\partial_{x_i z_i z_i} - \frac{2}{3}\partial_{x_i x_i x_i}\right) & 0 & -\left(\frac{5}{3}\partial_{x_i x_i z_i}\right) \\ 0 & \left(\partial_{x_i x_i x_i} + \partial_{x_i z_i z_i}\right) & 0 \\ -\left(\frac{5}{3}\partial_{x_i x_i z_i}\right) & 0 & \left(\partial_{x_i x_i x_i} - \frac{2}{3}\partial_{x_i z_i z_i}\right) \end{bmatrix} g_2(\mathbf{r}_i;\sigma) \\
\mathbf{H}_{f_i,z_i}(\mathbf{r}_i) &= f * \begin{bmatrix} \left(\partial_{z_i z_i z_i} - \frac{2}{3}\partial_{x_i x_i z_i}\right) & 0 & -\left(\frac{5}{3}\partial_{x_i z_i z_i}\right) \\ 0 & \left(\partial_{x_i x_i z_i} + \partial_{z_i z_i z_i}\right) & 0 \\ -\left(\frac{5}{3}\partial_{x_i z_i z_i}\right) & 0 & \left(\partial_{x_i x_i z_i} - \frac{2}{3}\partial_{z_i z_i z_i}\right) \end{bmatrix} g_2(\mathbf{r}_i;\sigma)
\end{aligned}$$

Plugging (42) into the integral I_1 in (43) , we get

$$I_2 = \sum_{i=0}^{N-1} \left[\mathbf{P}_i(0,0), \mathbf{P}_i(1,0)\right] \begin{bmatrix} \int_0^M \left(d\mathbf{r}^T(t) \mathbf{R}_i^T \mathbf{H}_{f_i,x_i}(\mathbf{r}_i(t)) \mathbf{R}_i d\mathbf{r}(t) \beta_p^n(t-k) \right) dt \\ \int_0^M \left(d\mathbf{r}^T(t) \mathbf{R}_i^T \mathbf{H}_{f_i,z_i}(\mathbf{r}_i(t)) \mathbf{R}_i d\mathbf{r}(t) \beta_p^n(t-k) \right) dt \end{bmatrix}$$

Thus, we get

$$\begin{aligned}
\begin{bmatrix} \frac{\partial}{\partial c_x(k)} \\ \frac{\partial}{\partial c_y(k)} \\ \frac{\partial}{\partial c_z(k)} \end{bmatrix} E_{\text{image}} &= -2 \int_0^M \left(d\mathbf{r}(t)^T \mathbf{H}_{\mathrm{3D}}(\mathbf{r}(t)) \beta_p^{n\prime}(t-k) \right) dt \\
&\quad -2 \sum_{i=0}^{N-1} \mathbf{P}_i^T \begin{bmatrix} \int_0^M \left(d\mathbf{r}^T(t) \mathbf{R}_i^T \mathbf{H}_{f_i,x_i}(\mathbf{r}_i(t)) \mathbf{R}_i d\mathbf{r}(t) \beta_p^n(t-k) \right) dt \\ \int_0^M \left(d\mathbf{r}^T(t) \mathbf{R}_i^T \mathbf{H}_{f_i,z_i}(\mathbf{r}(t)) \mathbf{R}_i d\mathbf{r}(t) \beta_p^n(t-k) \right) dt \end{bmatrix} \qquad (43)
\end{aligned}$$

The evaluation of the matrices $\mathbf{H}_{f_i,x_i}$ and $\mathbf{H}_{f_i,y_i}$ necessitates the computation of the quantities $(f_i * g_{x_i x_i x_i})$, $(f_i * g_{x_i x_i y_i})$, $(f_i * g_{x_i y_i y_i})$ and $(f_i * g_{y_i y_i y_i})$ for each micrograph. For the first term in (43), we require the matrix $\mathbf{H}_{f_i}$, which in-turn needs the quantities $(f_i * g_{x_i x_i})$, $(f_i * g_{x_i y_i})$ and $(f_i * g_{y_i y_i})$. Note that all these quantities involve the convolution of f with the second and third order partial derivatives of the 2-D Gaussian; they can be pre-computed efficiently using separable linear filtering. We discretize the integrals

for their evaluation. Thanks to the steerable implementation, the criterion and its partial derivatives can be computed exactly and efficiently.

B. Partial derivatives of the internal energy

The internal energy term can be re-written as

$$\int_0^M \left|\mathbf{r}''(t)\right|^2 dt = \int_0^M \left(\left|x''(t)\right|^2 + \left|y''(t)\right|^2 + \left|z''(t)\right|^2\right) dt \tag{44}$$

We now consider the term $\int_0^M |x''(t)|^2 dt$ and simplify it as follows:

$$\begin{aligned}\int_0^M \left|x''(t)\right|^2 dt &= \sum_{k=0}^{M-1}\sum_{l=0}^{M-1} c_x(k)\, c_x(l) \int_0^M \beta_p^{n\prime\prime}(t-k)\,\beta_p^{n\prime\prime}(t-l)\, dt \\ &= \sum_{l=0}^{M-1}\sum_{k=-\infty}^{\infty} c_x(k)\, c_x(l) \underbrace{\int_{-\infty}^{\infty} \beta^{n\prime\prime}(t-k)\,\beta_p^{n\prime\prime}(t-l)\, dt}_{q(k-l)}\end{aligned} \tag{45}$$

In the last step, we have used the periodicity of $\beta_p^{n\prime\prime}(t-l)$ to extend the integral from $-\infty$ to ∞ and have transferred the periodicity of $\beta_p^{n\prime\prime}(t-k)$ to the coefficient sequence c_k. Thanks to the curve representation using cubic B-splines, the sequence $q(k)$ is finitely supported and can be exactly computed. Thus, we obtain the partial derivatives of the internal energy term as

$$\begin{bmatrix} \frac{\partial}{\partial c_x(k)} \\ \frac{\partial}{\partial c_y(k)} \\ \frac{\partial}{\partial c_z(k)} \end{bmatrix} E_{\text{int}} = \sum_{k=-\infty}^{\infty} \begin{bmatrix} c_x(k) \\ c_y(k) \\ c_z(k) \end{bmatrix} q(k-l)\,.$$

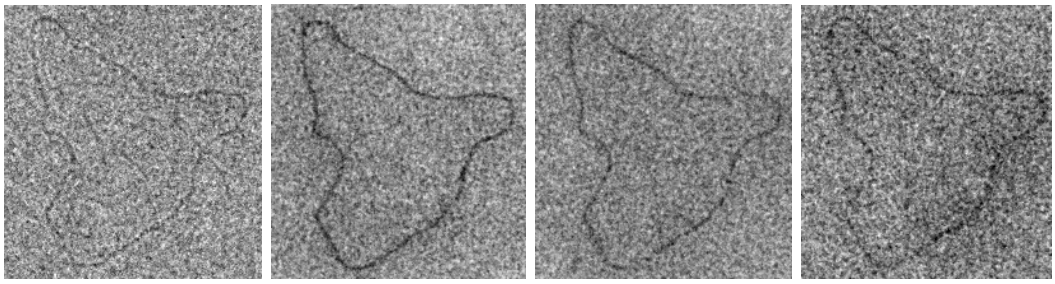
The above equation amounts to a simple filtering of the coefficient sequence by the filter $q(n)$, assuming periodic boundary conditions.

C. Partial derivatives of the constraint energy

1) Length constraint: Differentiating (38) with respect to $c_x(k)$ gives

$$\begin{aligned}\frac{\partial}{\partial c_x(k)} E_{\text{const}} &= 2\underbrace{\left(\int_0^M \left|\mathbf{r}'(t)\right| dt - \text{Length}\right)}_{\text{Error}} \int_0^M \frac{x'(t)\,\beta_p^{n\prime}(t-k)}{\left|\mathbf{r}'(t)\right|} dt \\ &= 2\,\text{Error} \int_{-\infty}^{\infty} \frac{x'(t+k)\,\beta^{n\prime}(t)}{\left|\mathbf{r}'(t+k)\right|} dt\end{aligned} \tag{46}$$

Here, Error is the difference between the current length of the curve and the expected one; the partial derivatives of the constraint energy is zero when the current length is the same as the expected one. The partial derivatives $\frac{\partial}{\partial c_y(k)} E_{\text{const}}$ and $\frac{\partial}{\partial c_z(k)} E_{\text{const}}$ is computed in a similar fashion.



(a) original image (b) STD=40 (c) STD=55 (d) STD=70

Fig. 6. Simulated data for different noise standard deviations. The projection data is corrupted with gaussian noise of specified standard deviations and then convolved with a filter whose magnitude response is matched to the power spectral density of the micrograph noise. For real micrographs, the noise standard deviation is in the range 50-60.

Evolving the curve with this term alone will cause its length to decrease or increase, (depending on the sign of Error) until $\text{Error} = 0$. Note that the integral (46) is limited over the support of the spline function. We discretize the integral for its evaluation.

2) Point constraint: Computing the partial derivatives of (39) in all generality would give a very complicated expression. To make the problem more tractable and to reduce its computational complexity, we make the assumption that the optimal constraint locations, $(t_i;\ i = 0 \dots N_{\text{c}} - 1)$, are known. In this case, (39) gets simplified to

$$E_{\text{c}} = \sum_{i=0}^{N_{\text{c}}-1} \left| \mathbf{r}\left(t_i\right) - \mathbf{r}_{\text{c},i} \right|^2, \tag{47}$$

and its partial derivatives are given by

$$\begin{bmatrix} \partial E_{\text{c}}/\partial c_{x,k} \\ \partial E_{\text{c}}/\partial c_{y,k} \end{bmatrix} = \sum_{i=0}^{N_{\text{c}}-1} \left(\begin{bmatrix} x_{\text{c},i} \\ y_{\text{c},i} \end{bmatrix} - \begin{bmatrix} x\left(t_i\right) \\ y\left(t_i\right) \end{bmatrix} \right) \beta^n\left(t_i - k\right) \tag{48}$$

Using the finite support of the scaling functions, we limit the sum to the relevant indices (we need to evaluate it only for $\{i |\ 0 < (t_i - k) < N_{\text{c}}\}$). We resort to a two-step strategy, where the snake is evolved using the above formulas for the derivatives, for a given t_i. The optimal parameters t_i are then re-estimated within the loop as:

$$t_i = \arg \min_{t \in [0,M]} \left| \mathbf{r}\left(t\right) - \mathbf{r}_{\text{c},i} \right|;\ \ i = 0 \dots N_{\text{c}} - 1 \tag{49}$$

VI. Experiments

In this section, we evaluate the performance of the snake algorithm using simulated images (since we do not have the ground truth available with the real data). We consider a 3-D curve (specified by a user)

Noise STD	20	40	55	70	85	User input
3-D errors	1.3158 ± 0.3804	2.0780 ± 0.6532	2.2590 ± 0.5990	2.8530 ± 0.8340	4.0182 ± 1.3940	6.5430 ± 1.6570
Inplane errors	0.6656 ± 0.0604	0.7244 ± 0.0908	0.7888 ± 0.1245	0.8697 ± 0.2341	1.2979 ± 0.5479	3.0925 ± 0.5759

TABLE II

COMPARISON WITH THE REFERENCE CURVE FOR SIMULATED MICROGRAPHS AT DIFFERENT NOISE LEVELS. NOTE THAT THE IN-PLANE ERRORS REMAIN AT THE SUB-PIXEL RANGE FOR MOST CASES.

and imitate the image formation process. We generate a 3-D volume from the curve samples (we assume the filament to have a Gaussian cross-section) and project the volume onto the image planes. To account for the noise and the contrast transfer function of the microscope, we add Gaussian white noise of a specified standard deviation to the images and then filter[9] them. Some examples of simulated data and the corresponding real micrographs are shown in Fig. 6. We use the snake algorithm to reconstruct the shape which is then compared with the ground truth.

To compare two 3-D curves $\mathcal{C}_a$ and $\mathcal{C}_b$, we choose the error metric

$$\mathcal{D}\left(\mathcal{C}_a, \mathcal{C}_b\right) = \frac{1}{2}\left(\frac{1}{M_a}\int_0^{M_a} D\left(\mathbf{r}_a\left(t\right), \mathcal{C}_b\right) dt + \frac{1}{M_b}\int_0^{M_b} D\left(\mathbf{r}_b\left(t\right), \mathcal{C}_a\right) dt\right), \tag{50}$$

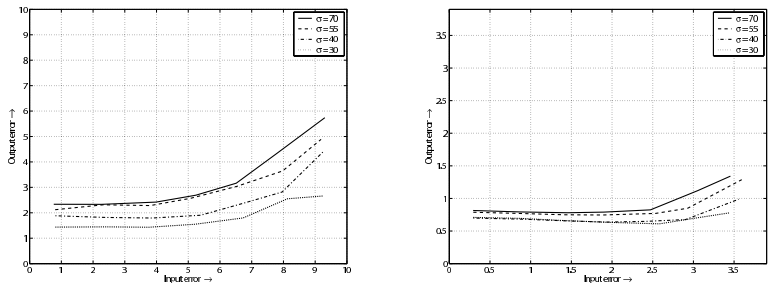
where $\mathcal{C}_a \equiv \ \mathbf{r}_a\left(t\right);\ t \in [0, M_a]$ and $\mathcal{C}_b \equiv \ \mathbf{r}_b\left(t\right);\ t \in [0, M_b]$. The distance between a point $\mathbf{r}_a\left(t\right)$ and a curve $\mathcal{C}_b$ (denoted in the above equation as $D\left(\mathbf{r}_a\left(t\right), \mathcal{C}_b\right)$) is defined as the distance between $\mathbf{r}_a\left(t\right)$ and the closest point on $\mathcal{C}_b$:

$$D\left(\mathbf{r}_a\left(t\right), \mathcal{C}_b\right) = \min_{t \in [0, M]} \left|\left|\mathbf{r}_a\left(t\right) - \mathbf{r}_b\left(t\right)\right|\right| \tag{51}$$

We evaluate the distance metric by discretizing both curves.

The reconstructions were performed on 6 stereo pairs, starting with user specified initialization. Two independent users performed 5 trials each for every stereo pair; they were instructed to use the graphical interface to specify of the order of 10 corresponding pairs of points per curve which would then be interpolated to provide a good starting point for the algorithm (user input). The results were compared with the corresponding reference curves to obtain the absolute errors in 3-D as well as on the micrograph planes. The average errors are given in Table VI. Note that the in-plane reconstruction errors are not very sensitive to micrograph noise; they remain in the sub-pixel range for reasonable noise levels. In

[9]We use a filter whose magnitude response is the square root of the power spectral density of the micrograph noise and a constant phase response.



(a) 3-D error (b) In-plane error

Fig. 7. Study of the sensitivity of the algorithm to initialization. The algorithm is initialized using random perturbations of the actual curve projections and the reconstructions are compared with the original data. We used the standard deviation of the 3-D template as 2.5 pixels. The plots shows the reconstruction errors as a function of the input errors for different levels of image noise (denoted by σ — the standard deviation of the noise added). Note that the reconstructions are robust to initial conditions; there is a factor of 3-4 decrease in the inplane errors while the 3-D errors are lower approximately by a factor of 2.

comparison, the 3-D errors are more sensitive to noise but are significantly lower than the manual tracings. Also note that the standard deviation of the errors are much lower than the ones associated with the manual tracings; this indicates that the reconstruction process is quite reproducible.

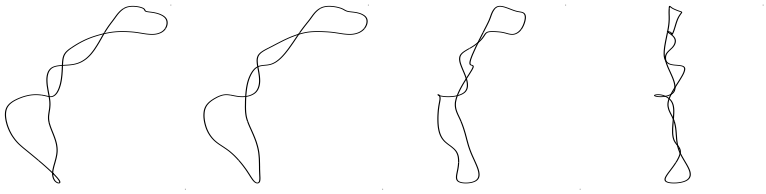
To study the sensitivity of the algorithm to initialization, we randomly perturbed the initial curve projections by adding correlated noise to the reference B-spline coefficients[10]. The results of the comparisons for input perturbations of different standard deviations are shown in Fig. 7. Note that in this case too, the in-plane errors remain at sub-pixel level irrespective of the initialization. The curves are almost flat indicating the low sensitivity of the algorithm to initial conditions.

Some examples of 3-D reconstructions obtained by applying our algorithm to real data are shown in Fig. 8—Fig. 10. This illustrate the wide range of DNA configurations that may occur in nature as well as the difficulty of the problem.

VII. Conclusion

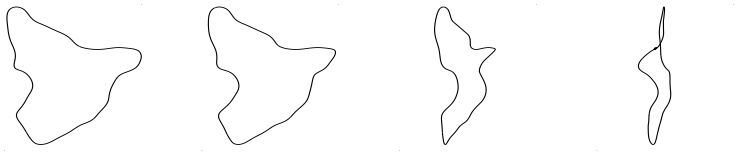
We have presented a carefully engineered solution for the 3-D shape estimation of DNA molecules from stereo cryo-electron micrographs. We used a global 3-D model for the DNA filament and optimized its parameters such that its 2-D orthogonal projections matched with the micrographs. Since a direct implementation of this algorithm is computationally expensive, we approximated the model locally using elongated blobs.

[10]To simulate the user initialization.



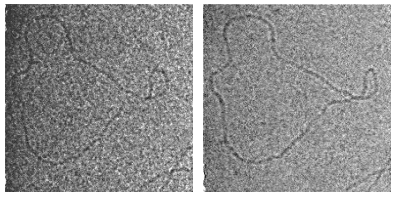
(a) -15° (b) 15° (c) 60° (d) 90°

Fig. 8. Reconstructed filaments for the micrograph pair in Fig. 1 at different viewing angles around the vertical axis. (a) and (b) correspond to the left and the right micrographs in Fig. 1. The algorithm started from the user-specified initialization and generated the reconstructions in 6.7 seconds on a 667 MHz Macintosh G4. The values of the parameters μ_{int} and μ_{const} were 0.4 and 0.15 respectively.

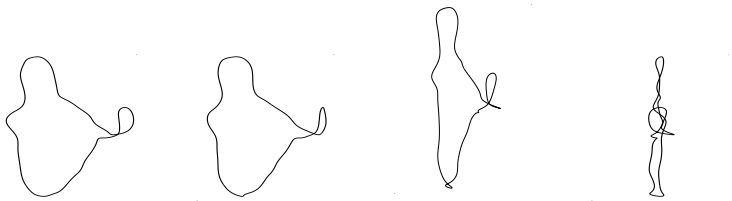


(a) -15° (b) 15° (c) 60° (d) 90°

Fig. 9. Reconstructed filaments for the micrograph Fig. 6 at different viewing angles around the vertical axis. (a) to the micrograph in Fig. 6-a . The algorithm started from the user-specified initialization and generated the reconstructions in 5.3 seconds on a 667 MHz Macintosh G4.The values of the parameters μ_{int} and μ_{const} were 0.4 and 0.15 respectively.



(a) -15° (b) 15°



(c) -15° (d) 15° (e) 60° (f) 90°

Fig. 10. Reconstructed filaments at different viewing angles around the vertical axis. (c) and (d) correspond to the left and the right micrographs in (a) and (b). The algorithm started from the user-specified initialization and generated the reconstructions in 7.1 seconds on a 667 MHz Macintosh G4. The values of the parameters μ_{int} and μ_{const} were 0.4 and 0.15 respectively.

To solve the local detection problem, we introduced the concept of projection-steerablity. Specifically, we derived a projection-steerable elongated blob template whose 2-D projections can be represented as a linear combination of few basis functions. We also developed an efficient algorithm for obtaining the likeliness of such a blob with a specific orientation at a certain point in 3-D space.

We used a 3-D B-spline curve model for the representation of the skeleton of the global DNA model. We showed that the B-spline representation is optimal for the representation of smooth 3-D curves, if described in the constant arc-length parameterization. Using the constant arc-length assumption, we obtained a simple expression for an internal energy term that penalizes the average curvature magnitude.

We used a conjugate gradients algorithm for the optimization of the curve parameters. Thanks to the projection-steerable blob detection algorithm and curve representation using compactly supported B-spline functions, all the directional derivatives of the snake energies are computed exactly and efficiently.

ACKNOWLEDGEMENTS

We would like to that Prof. Jacques Dubochet, Catherine El-Bez and Erik Larquet of the LAU, UNIL for the active discussions and the data. This work was supported by the Beckman Foundation and the Swiss National Science Foundation under grant 2100-053540.

APPENDIX

A. *Design of the 3-D projection-steerable elongated blob*

We now derive a 3-D detector in $V_{3\mathrm{D}}$ that is optimized for the detection of a filament with a specific orientation (say along the x axis). Since the template is rotation-steerable by construction, we can steer it shape to any orientation exactly. An arbitrary function in $V_{3\mathrm{D}}$ is given by

$$h = a_0\,\partial_{xx}\,g_3 + a_1\,\partial_{yy}\,g_3 + a_2\,\partial_{zz}\,g_3 + a_3\,\partial_{xy}\,g_3 + a_4\,\partial_{xz}\,g_3 + a_5\,\partial_{yz}\,g_3 \tag{52}$$

Since a 1-D ridge oriented along the x axis is an even function along the axes, the terms $\partial_{xy}\,g_3$, $\partial_{xz}\,g_3$ and $\partial_{yz}\,g_3$ (they are odd functions) will not contribute to the ridge signal. Hence, we set a_3, a_4 and a_5 to zero.

We like to have a detector that is elongated along the x axis and narrow along the y and the z axis. The elongation along the axes can be measured by the magnitude of the second derivatives of h at the origin. For the detector to be maximally elongated along the x axis, we set $\partial_{xx}h|_{0,0,0}$ to zero:

$$\partial_{xx}h\,(\mathbf{r})\,|_{\mathbf{r}=(0,0,0)} = \frac{1}{\sigma^4}\,(3a_0 + a_1 + a_2) = 0 \tag{53}$$

For the filter to be narrow along y and the z axes, we have to maximize the quantities

$$\partial_{yy} h(\mathbf{r})\,|_{\mathbf{r}=(0,0,0)} = (a_0 + 3a_1 + a_2) \tag{54}$$

$$\partial_{zz} h(\mathbf{r})\,|_{\mathbf{r}=(0,0,0)} = (a_0 + a_1 + 3a_2) \tag{55}$$

We maximize $\partial_{yy}h|_{0,0,0} + \partial_{zz}h|_{0,0,0}$ subject to (53) and the unit energy constraint:

$$\|h(\mathbf{r})\|^2 = 3a_0^2 + 3a_1^2 + 3a_2^2 + 2a_0a_1 + 2a_0a_2 + 2a_1a_2 = 1 \tag{56}$$

We have removed the constants in all the above equations to simplify the formulas. The constants will not affect the shape of the detector. From (53), we have $a_1 + a_2 = -3a_0$. With this condition, we have to maximize

$$\partial_{yy} h(\mathbf{r})\,|_{\mathbf{r}=(0,0,0)} + \partial_{zz} h(\mathbf{r})\,|_{\mathbf{r}=(0,0,0)} = (2a_0 + 4a_1 + 4a_2) \tag{57}$$

$$= -10a_0 \tag{58}$$

subject to (56):

$$\|h(\mathbf{r})\|^2 = 3a_0^2 + 3\underbrace{(a_1^2 + a_2^2)}_{\frac{(a_1+a_2)^2+(a_1-a_2)^2}{2}} + 2a_0\underbrace{(a_1 + a_2)}_{-3a_0} + \underbrace{2a_1a_2}_{\frac{(a_1+a_2)^2-(a_1-a_2)^2}{2}} \tag{59}$$

$$= 15a_0^2 + (a_1 - a_2)^2 = 1 \tag{60}$$

Thus the solution is $a_0 = -1/\sqrt{15}$ and $a_1 = a_2$, which simplifies to $a_1 = 3/2\sqrt{15}$, $a_2 = 3/2\sqrt{15}$; the expression for the detector is given by

$$h(\mathbf{r}) = \sqrt{\frac{3}{20}}\left(\partial_{yy}\, g_3(\mathbf{r},\sigma) + \partial_{zz}\, g_3(\mathbf{r},\sigma) - \frac{2}{3}\,\partial_{xx}\, g_3(\mathbf{r},\sigma)\right) \tag{61}$$

B. Projection of the optimal 3-D detector

The expression of the optimal filament detector, oriented along along $\mathbf{v}$, is given by

$$h_{\mathbf{v}}(\mathbf{r}) = \underbrace{g_{xx}(\mathbf{r};\sigma) + g_{yy}(\mathbf{r};\sigma) + g_{zz}(\mathbf{r});\sigma}_{\text{Laplacian of } g(\mathbf{r})} - \frac{5}{3}\, g_{\mathbf{vv}}(\mathbf{r};\sigma)\,. \tag{62}$$

Note that in this case $\mathbf{A} = \mathbf{I}_3 - \frac{5}{3}\,\mathbf{v}\mathbf{v}^T$, where $\mathbf{I}_3$ is the 3x3 identity matrix.

Now, using (22), we obtain the corresponding $\mathbf{B}_i$ matrix as

$$\mathbf{B}_i = \underbrace{\mathbf{P}_i^T\mathbf{P}_i}_{\mathbf{I}_2} - \frac{5}{3}\underbrace{(\mathbf{P}_i\mathbf{v})\,(\mathbf{P}_i\mathbf{v})^T}_{\mathbf{v}_i\,\mathbf{v}_i^T}, \tag{63}$$

which implies that the Fourier transform of the 2-D projection of the template is given by

$$\hat{h}_i(\boldsymbol{\omega}_i) = -\left(\|\boldsymbol{\omega}_i\|^2 - \frac{5}{3}\langle \boldsymbol{\omega}_i, \mathbf{v}_i\rangle^2\right) g\left(\boldsymbol{\omega}_i; \sigma^{-1}\right). \tag{64}$$

The above expression can also be written as

$$\hat{h}_i(\boldsymbol{\omega}_i) = -\left(\|\boldsymbol{\omega}_i\|^2 - \frac{5}{3}\,\mathbf{v}_i^T \underbrace{\begin{bmatrix} \omega_{x_i}^2 & \omega_{x_i}\omega_{z_i} \\ \omega_{x_i}\omega_{z_i} & \omega_{z_i}^2 \end{bmatrix}}_{\boldsymbol{\omega}_i\,\boldsymbol{\omega}_i^T} \mathbf{v}_i\right) g\left(\boldsymbol{\omega}_i; \sigma^{-1}\right). \tag{65}$$

Since $\|\mathbf{v}_i\|^2 + v_{y_i}^2 = \|v\|^2 = 1$, we rewrite this expression as

$$\begin{aligned}
\hat{h}_i(\boldsymbol{\omega}_i) &= -\left(\|\boldsymbol{\omega}_i\|^2 v_{y_i}^2 + \mathbf{v}_i^T \begin{bmatrix} \omega_{x_i}^2 + \omega_{z_i}^2 & 0 \\ 0 & \omega_{x_i}^2 + \omega_{z_i}^2 \end{bmatrix} \mathbf{v}_i - \frac{5}{3}\mathbf{v}_i^T \begin{bmatrix} \omega_{x_i}^2 & \omega_{x_i}\omega_{z_i} \\ \omega_{x_i}\omega_{z_i} & \omega_{z_i}^2 \end{bmatrix} \mathbf{v}_i\right) g\left(\boldsymbol{\omega}_i; \sigma^{-1}\right) \\
&= -\left(\|\boldsymbol{\omega}_i\|^2 v_{y_i}^2 - \mathbf{v}_i^T \begin{bmatrix} \omega_{z_i}^2 - \frac{2}{3}\omega_{x_i}^2 & -\frac{5}{3}\omega_{x_i}\omega_{z_i} \\ -\frac{5}{3}\omega_{x_i}\omega_{z_i} & \omega_{x_i}^2 - \frac{2}{3}\omega_{z_i}^2 \end{bmatrix} \mathbf{v}_i\right) g\left(\boldsymbol{\omega}_i; \sigma^{-1}\right).
\end{aligned} \tag{66}$$

which is then modified to

$$\hat{h}_i(\boldsymbol{\omega}_i) = -\left(\mathbf{v}^T \mathbf{R}_i \begin{bmatrix} \omega_{z_i}^2 - \frac{2}{3}\omega_{x_i}^2 & 0 & -\frac{5}{3}\omega_{x_i}\omega_{z_i} \\ 0 & \omega_{x_i}^2 + \omega_{z_i}^2 & 0 \\ -\frac{4}{3}\omega_{x_i}\omega_{z_i} & 0 & \omega_{x_i}^2 - \frac{2}{3}\omega_{z_i}^2 \end{bmatrix} \mathbf{R}_i \mathbf{v}\right) g\left(\boldsymbol{\omega}_i; \sigma^{-1}\right), \tag{67}$$

where $\mathbf{R}_i$ is the rotation matrix given by (10) and (**??**). Finally, by computing the inverse Fourier transform, we get

$$\mathcal{P}_i\left(h_{\mathbf{v}}(\mathbf{r})\right) = \mathbf{v}^T \mathbf{R}_i \underbrace{\left(\begin{bmatrix} \left(\partial_{z_i z_i} - \frac{2}{3}\partial_{x_i x_i}\right) & 0 & -\left(\frac{5}{3}\partial_{x_i z_i}\right) \\ 0 & \left(\partial_{x_i x_i} + \partial_{z_i z_i}\right) & 0 \\ -\left(\frac{5}{3}\partial_{x_i z_i}\right) & 0 & \left(\partial_{z_i z_i} - \frac{2}{3}\partial_{x_i x_i}\right) \end{bmatrix} g_3\left(\mathbf{r}_i; \sigma\right)\right)}_{G(\mathbf{r}_i;\sigma)} \mathbf{R}_i\,\mathbf{v}. \tag{68}$$

REFERENCES

[1] J. Dubochet, M. Adrian, J. Chang, J. C. Homo, J. Lepault, A. W. McDowall, and P. Schultz, "Cryo-electron microscopy of vitrified specimens," *Quarterly review of Biophysics*, vol. 21(2), pp. 129–228, 1988.

[2] J. Dubochet, M. Adrian, I. Dustin, P. Furrer, and A. Stasiak, "Cryoelectron microscopy of DNA molecules in solution," *Methods Enzymol.*, vol. 211, pp. 507–518, 1992.

[3] M. Adrian, B. H. Bordier, W. Wahli, A. Stasiak, and J. Dubochet, "Direct visualization of supercoiled DNA molecules in solution," *The EMBO journal*, vol. 9(13), pp. 4551–4554, 1990.

[4] I. Dustin, P. Furrer, A. Stasiak, J. Dubochet, J. Langowski, and E. Egelman, "Spatial visualization of DNA in solution," *Journal of Structural Biology*, vol. 107, pp. 15–21, 1991.

[5] F. P. I. Margalef, P. Furrer, M. Kunt, and J. Dubochet, "The flying cylinder: A new algorithm for filament recognition in noisy stereo images," *Journal of Structural Biology*, vol. 116, pp. 25–29, 1996.

[6] J. Bednar, P. Furrer, V. Katritch, A. Stasiak, J. Dubochet, and A. Stasiak, "Determination of DNA persistence length by cryo-electron microscopy," *Journal of Molecular Biology*, vol. 254, pp. 579–594, 1995.

[7] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891–906, 1991.

[8] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. Heeger, "Shiftable multiscale transforms," *IEEE Transactions on Information Theory*, vol. 38, pp. 587–607, 1992.

[9] P. C Teo and Y. Hel-Or, "Design of multi-parameter steerable functions using cascade-basis reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 552–556, 1999.

[10] S. Menet, P. Saint-Mark, and G. Medioni, "B-snakes: implementation and application to stereo," in *Image Understanding workshop*, 1990, pp. 720–726.

[11] P. Brigger, J. Hoeg, and M. Unser, "B-spline snakes: A flexible tool for parametric contour detection," *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1484–1496, 2000.

[12] M. Jacob, T. Blu, and M. Unser, "Efficient energies and algorithms for parametric snakes," *IEEE Transactions on Image Processing*, vol. 13, no. 9, 2004.

[13] R. H. Bartels, J. C. Bealty, J. C. Beatty, and B. A. Barsky, *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*, Morgan Kauffmann Publishers Inc, 1987.

[14] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, vol. 16, no. 6, pp. 22–38, 1999.

[15] A. Klug, "From macromolecules to biological assembiles," *Nobel Lectures, Chemistry 1981-90, World Scientific Publishing*, 1992.

[16] R.A Crowther, D.J. DeRosier, and A. Klug, "The reconstruction of a three dimensional structure from projections and its application to electron microscopy," *Proc. Royal. Soc. London*, vol. 317, pp. 319–340, 1970.

[17] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*, IEEE Press, 1988.

[18] M. Jacob and M. Unser, "Design of steerable filters for feature detection using canny-like criterion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1007–1019, 2004.

[19] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321–332, 1988.

[20] A. K. Jain, Y. Zhong, and M. P. D. Jolly, "Deformable template models: A review," *Signal Processing*, vol. 76, pp. 109–129, 1998.

[21] C. Canero, P. Radeva, R. Toledo, J. J. Villanueva, and J. Mauri, "3-D curve reconstruction by biplane snakes," in *International Conference on Pattern Recognition*, 2000, pp. 4563–4567.

[22] C. Canero, F. Vilarino, J. Mauri, and P. Radeva, "Predictive undistortion model and 3-D reconstruction by biplane snakes," *IEEE Transactions on Medical Imaging*, vol. 21, pp. 1188–1201, 2002.

[23] D. J. Struik, *Lectures on classical differential geometry*, Dover Publications, 2nd edition, 1988.

[24] I. J. Schoenberg, "Spline functions and the problem of graduation," *Proc. Nat. Acad. Sci.*, vol. 52, pp. 947–950, 1964.

[25] T. Blu and M. Unser, "Quantitative Fourier analysis of approximation techniques: Part II—Wavelets," *IEEE Transactions on Signal Processing*, vol. 47, no. 10, pp. 2783–2795, 1999.