Hindawi
Scanning
Volume 2019, Article ID 4235865, 14 pages
https://doi.org/10.1155/2019/4235865

Research Article

Resolution-Free Accurate DNA Contour Length Estimation from Atomic Force Microscopy Images

Peter I. Chang and Ming-Chi Hsaio

Mechanical Engineering, National Taiwan University of Science and Technology, Taiwan

Correspondence should be addressed to Peter I. Chang; itchang@mail.ntust.edu.tw

Received 27 April 2018; Revised 28 January 2019; Accepted 27 February 2019; Published 9 June 2019

Academic Editor: Daniele Passeri

This research presented an accurate and efficient contour length estimation method developed for DNA digital curves acquired from Atomic Force Microscopy (AFM) images. This automation method is calibrated against different AFM resolutions and ideal to be extended to all different kinds of biopolymer samples, encompassing all different sample stiffnesses. The methodology considers the digital curve local geometric relationship, as these digital shape segments and pixel connections represent the actual morphology of the biopolymer sample as it is being imaged from the AFM scanning. In order to incorporate the true local geometry relationship that is embedded in the continuous form of the original sample, one needs to find this geometry counterpart in the digitized image. This counterpart is realized by taking the skeleton backbone of the sample contour and by using these digitized pixels' connection relationship to find its local shape representation. In this research, one uses the 8-connect Freeman Chain Code (CC) to describe the directional connection between DNA image pixels, in order to account for the local shapes of four connected pixels. The result is a novel shape number (SN) system derived from CC, which is a fully automated algorithm that can be applied to DNA samples of any length for accurate estimation, with efficient computational cost. This shape-wise consideration is weighted to modify the local length with great precision, accounting for all the different morphologies of the biopolymer sample, and resulted with accurate length estimation, as the error falls below 0.07%, an order of magnitude improvement compared to previous findings.

1. Introduction

The Atomic Force Microscopy (AFM) system has the ability to probe samples at the nanometer scale, owing to its ability in sensing the sample surface to resolve force interaction at the pico-Newton level [1]. This feature makes AFM systems a useful imaging device in the field of nanotechnology, molecular biology, and many others. It is well known in AFM's biological application to image biopolymers thanks to its ability to image in liquid, the biopolymer's natural environment [2].

One very interesting characteristic is the length of a single DNA strand, denoted as l_c. This contour length can be applied to identify genome editing results and other application outcomes [3]. And accuracy in getting l_c correct is essential at this scale, as there is small room for error in genome editing, since one base-pair distance for DNA is only 0.34 nm. Thus, AFM images of DNA samples provide means for such l_c studies on accurate DNA length estimation, like the image illustrated here in Figure 1.

There are two ways in finding l_c from AFM images. One is by manual fitting, and the other is by automatic skeleton tracing with image processing. Fitting typically relies on human operators picking specific positions along the DNA contour by examination on the acquired image, which relies on the trained eye of a scholar to map out the contour length l_c [5, 6], as illustrated in Figure 2.

On the other hand, the automatic l_c estimation traces the DNA image along its backbone skeleton. This is done by thinning the strand image to its median position from the

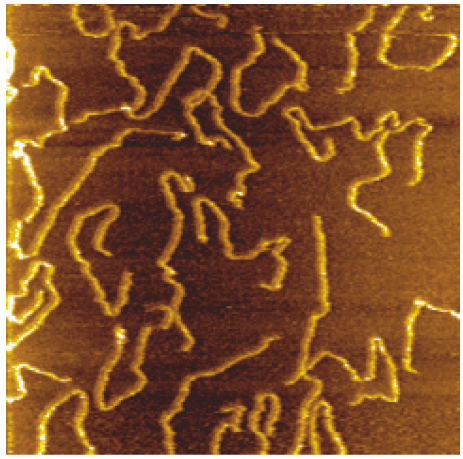


Figure 1: DNA image acquired by the atomic force microscopy system [4].

overall acquired outline and retaining only the skeleton of the DNA, as is illustrated in Figure 3.

The backbone extracted from the original AFM image is a *single-width* connected pixel arranged with the following rule: only one adjacent pixel is allowed to connect to the central pixel to form a continuous contour, either directly (horizontal or vertical) or diagonally, as is illustrated in Figure 4.

From this single width contour pixel arrangement, a continuous *chain code* (CC), defined as $C = c_1 c_2 \cdots c_n$, can be formed by tracing the connectedness of adjacent pixels from the skeleton's one end to the other, according to the 8-connect Freeman's eight directions [7].

Researchers have been using the Freeman CC to estimate l_c, by counting the number of even and odd occurrences along the DNA skeleton, which is to trace along the chain code, $C = c_i, i = 1 \sim n$, and tally up the occurrences of even number chain codes (n_e) as well as its odd occurrences (n_o).

Since the even chain code connects adjacent pixels directly (vertically/horizontally), and the odd CC connects diagonally, one estimates l_c first by finding the Euclidean length (norm) of all the pixel center connections and then multiply the pixel resolution r to find l_c. This is defined as the Freeman estimator $L_F = r(n_e + \sqrt{2n_o})$. [8].

However, L_F lacks the accuracy that is required in these microscopy systems. Thus, there are researches that made modifications to L_F. These include the Kupla estimator (L_K) and the corner estimator L_C. L_K modified the diagonal $\sqrt{2}$ values due to digital slope inclination, and L_C further accounts for tight turns geometrically. Thus, in the end L_K and L_C end up with different coefficients from L_F [9].

There were further researches to improve l_c accuracy. One research smooths out the digitized pixilation of the contour skeleton backbone and applied a spatial Fourier transform on the image. Through tuning the Gaussian filter in 2D, a smother l_c is estimated [10].

Other than modifying the pixel connection Euclidian length, another research modifies l_c by adjusting the pixel center coordinate representation x_p. A weight k is added to modify the coordinate location by considering the three consecutive points with $X_p = k(x_{p-1} - x_p) + x_p + k(x_{p+1} - x_p)$. This length estimator L_P calculates l_c according to the modified X_p [11].

Another l_c estimator is designed specifically for DNA strand samples, named L_{DNA}. This estimator introduced a nominal coefficient for different DNA lengths and is defined as $L_{DNA} = rC_f(n_e + \sqrt{2}n_o)$, where C_f is inversely calculated from simulated l_c data, so a table of C_f helps L_{DNA} to match the expected value of l_c [12].

More recently, a machine learning approach utilized a feature extraction to fit different cubic spline segment occurrences with the following: *horizontal*, *vertical*, *diagonal*, *perpendicular*, variating *height* and *thickness*, as defined by $\{n_{horz}, n_{vert}, n_{diag}, n_{perp}, n_{htcv}, n_{tkcv}\}$ [13]. This machine learning estimator L_{ML} is trained to generate coefficients considering the abovementioned feature from known DNA l_c.

A summary table in Table 1 provides a quick review of the abovementioned l_c estimators.

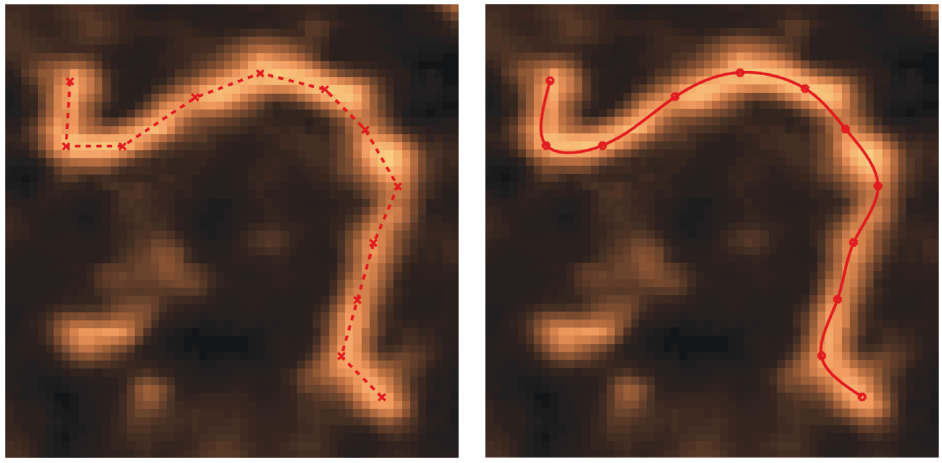
In this paper, the authors propose an estimator based on the DNA imaged contour shape, thus having the name *Shape* estimator L_S, where L_S is designed to be robust to image resolution and only uses minimal computational resource. This is achieved by considering the neighboring shape of the original two-pixel connection inspired from L_F, but as all the DNA local morphology shapes are considered for estimating l_c, the resultant accuracy is shown to improve by more than an order of magnitude.

Detailed methodology of the L_S estimator is explained in Section 2, starting from the general image preprocessing to the identification of twelve local 4-pixel segment configuration shapes. Then, the 12-shape correction coefficients $k_1 \sim k_{12}$ are calibrated in Section 3, with different resolutions considered. Finally, the l_c values for L_S are compared with L_{DNA} and L_F in Section 4.

2. Contour Length Estimation with Local Shape Consideration

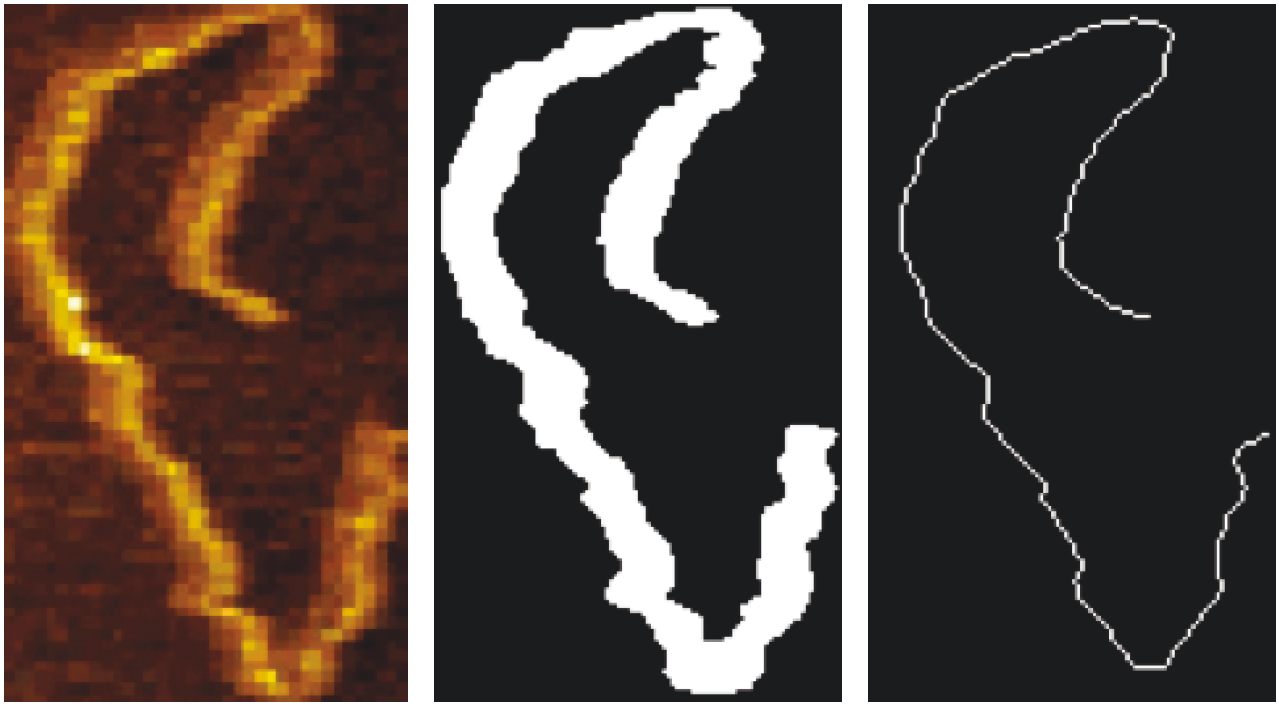
L_S estimation essentially takes into account the local shape considerations. As two neighboring pixels are connected together in this AFM image, the overall shape around the two connected pixels represents different local lengths as this DNA morphology is observed. In a tight turn; i.e., a "kink," this local length will certainly be longer then a smooth linear local profile.

Thus, L_S considers the two additional pixels extending from the center two-pixel connection and identifies the different 4-pixel segmented shapes surrounding along the DNA skeleton backbone. Then, L_S makes shape-corrected length adjustments, by multiplying the local shape's corresponding coefficient to adjust for the estimated l_c. It can be observed that the extension of this segmented elemental shape is not limited to 4 pixels, as with more pixels such as a 5-pixel segment can also be considered. However, due to the trade-off for computational cost and performance, this research investigates the L_S with 4-pixel elements.



(a) (b)

Figure 2: (a) Linear/straight fitting with selected points manually. (b) Cubic spline fit utilizing the same manual selective points on the AFM image.



(a) (b) (c)

Figure 3: Illustration of DNA image preprocess to extract contour skeleton backbone. (a) Original AFM image cut-out with a single DNA sample. (b) Binary image acquired by thresholding the original. (c) Skeleton extracted by a repetitive thinning process to produce a single-pixel width, connected contour.

2.1. Pixel Resolution and Image Preprocessing. A standard preprocess extracts the DNA image into the l_c's skeleton backbone, by thinning the DNA strand into the centerline of the biopolymer. This research's automatic image process is illustrated here in Figure 5.

First, the DNA image is prefiltered and mapped into a binary image with thresholding. Then, further, 2-D filters remove isolated pixel islands, ensuring that a single DNA contour is captured. And finally, an iterative debranch thinning morphology is applied to find the skeleton backbone that can be chain-coded [14].

It is well known that AFM systems have a tip broadening effect when imaging, which expands the DNA strand width to a larger value. A repeated thinning preprocess in average converges the single-width pixel contour, towards the mid-point of the DNA strand automatically, given an AFM image with enough resolution across the DNA width [15].

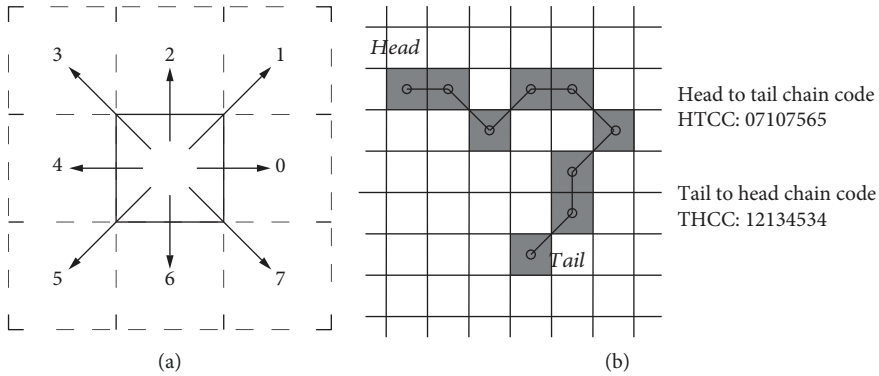


Figure 4: (a) 8-connect chain code connectivity from the center pixel, illustrating the connection code 0~7 associated according to the adjacent pixel location. (b) An example of a single-pixel width biopolymer skeleton backbone used for deriving the chain code. Starting from head to tail codes: 07107565 and tail to head codes: 12134534.

Table 1: Comparison of digital contour length methodology difference with biopolymer samples.

Author	Year	Estimator	Algorithm	Specific to DNA
Spisz et al. [8]	1998	Freeman estimator L_F	$L_F = r(n_e + \sqrt{2}n_o)$	
Rivetti and Codeluppi [9]	2001	Kupla L_K and corner L_C estimator	$L_K = r(0.948n_e + 1.343n_o)$, $L_C = r(0.980n_e + 1.406n_o - 0.091n_c)$	✓
Sanchez-Sevilla et al. [10]	2002	Curvature filtering	Coordinate transform to complex plane with Fourier transform	
Ficarra et al. [11]	2005	Pixel coordinate	Relocation of pixel representative coordinate X	
Rivetti [12]	2009	DNA estimator L_{DNA}	$L_{DNA} = rC_f(n_e + \sqrt{2}n_o)$, $C_f = \langle l_c \rangle / L_F$	✓
Sundstorm et al. [13]	2012	Machine learning	$L_{ML} = \sum \{n_{horz}, n_{vert}, n_{diag}, n_{perp}, n_{htcv}, n_{tkcv}\}$	✓

2.2. Identification of 4-Pixel Segment Shape Connectivity. Given the resultant single-width pixels $P = \{p_i, i = 0 \sim n\}$ for the contour's skeleton backbone, its CC $(C = c_i i = 1 \sim n)$ is coded from one end to the other. Note that this research utilizes the 8-connect chain code, resulting in integers ranging from $0 \sim 7$ for all c_i and that c_i is one off from p_i, as there are n connections between $n + 1$ pixels.

With the 4-pixel segment setup, there are up to a total of 64 ways (4^4) to connect the 4 pixels into single-pixel width arrangements. This research paper has fully outlined all the possibilities, and the full table of all 64 different single-width 4-pixel segments is arranged in Figures 6 and 7. They are arranged by the assigned $k_1 \sim k_{12}$ types, with all the same types grouped together.

It is clear that all the same types of k_j shape are grouped with the 4-pixel segment's mirror and rotational images. Take for example the k_8 shape, where the segment is rotated clockwise/counterclockwise for 90 degrees individually and mirrored on the y-axis, shown here in Figure 8.

Having these $k_1 \sim k_7$ segment shapes distinguished, the original inner 2-pixel connection's distance can now be corrected, by considering the outward extended 4-pixel segment shape. This would take into account the local geometric features according to its categorized shape. Since the skeleton backbone is composed of consecutive 4-pixel segments all along its contour, when tracing from one end to the other, this research makes sure that the L_S estimator identifies every 4-pixel segment to the *twelve* unique k_j shapes, as shown in Figure 9.

2.2.1. Chain Code, Shape Number, and Identifier. In order to identify a skeleton backbone's different 4-pixel segment shapes along the contour, this research utilizes its chain code, formed as a series of integer number, and developed a novel algorithm called the shape number (SN) identification, labeled S, and uses it to derive an exclusive identifier (ID) number for matching the abovementioned unique $k_1 \sim k_{12}$ shapes.

A typical CC collection, $C = \{c_1 c_2 c_3 \cdots c_{n-2}\, c_{n-1} c_n\}$, is a series of integers made from $0 \sim 7$, provided the $n + 1$ single-width skeleton backbone pixels $P = \{p_0 p_1 p_2 \cdot p_{n-1} p_n\}$. Note that C is one-off from P and that $p_i s$ is numbered from $0 \sim n$. This research emphasizes the general ability to distinguish any skeleton backbone, and while for any given backbone, it creates a set of two distinct CC for every skeleton, due to starting the connection from different ends of the pixel chain. The algorithm will demonstrate the ability to converge on the distinguished 4-pixel segment shapes.

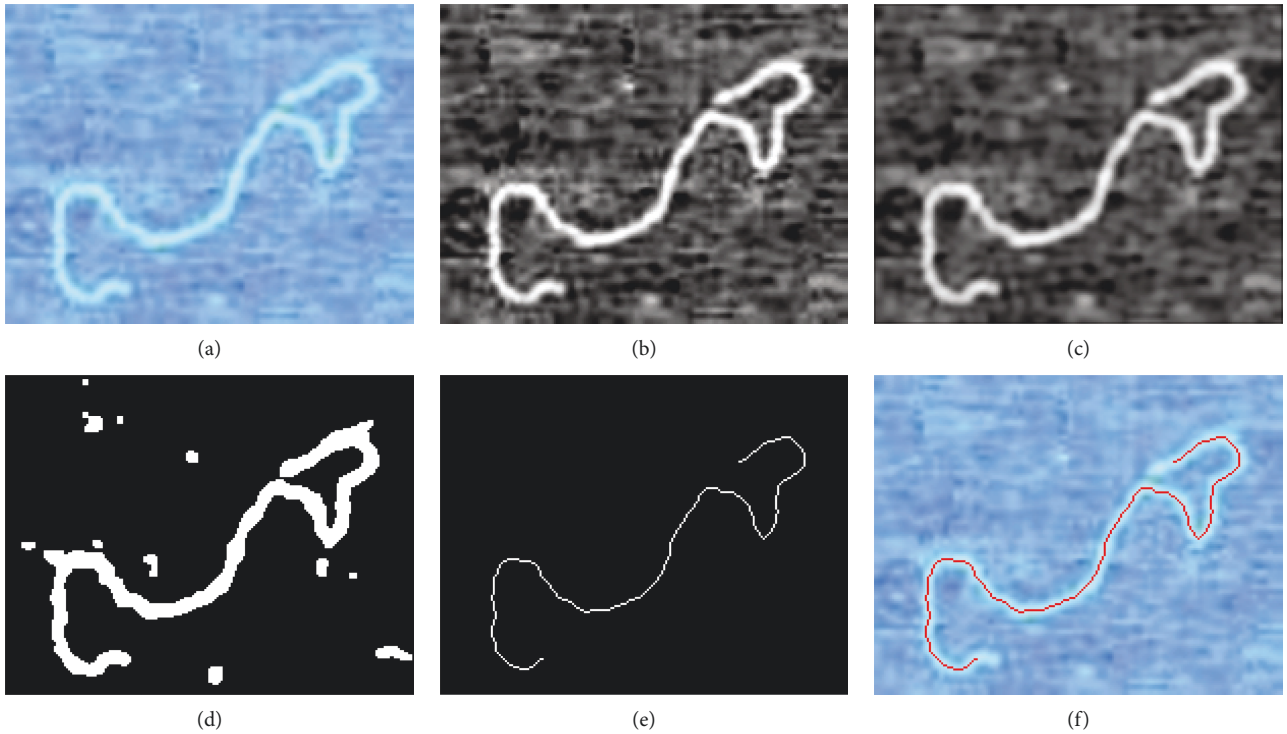


Figure 5: DNA image preprocess. (a) Original image, (b) binary transform, (c) image filtering, (d) image thresholding, (e) thinning and debranching and (f) final result compared with the original image.

As the algorithm needs to continuously identify the 4-pixel segments throughout the contour backbone, a rolling window starts from any end of C and collects the following n-segments (Figure 10)

$$\text{Seg} = \{\text{seg}_1, \text{seg}_2, \cdots, \text{seg}_i, \cdots, \text{seg}_n\}. \quad (1)$$

It is clear that each seg_i segment is comprised of three consecutive chain codes (c_{i-1}, c_i, c_{i+1}), since a 4-pixel segment consists of three connections. With the exception of the first and last segments of the contour skeleton, where there are not enough pixels to form a 4-pixel segment, thus the algorithm just takes the original two connecting pixels, i.e., the original c_1 or c_n. The pixel/geometric representation of a rolling window CC segmentation is demonstrated in the Figure 11.

Notice that the rolling window in C moves the 4-pixel segment consecutively from *Head* to *Tail*, and each segment can be coded as $\text{seg}_i = \{c_{i-1}, c_i, c_{i+1}\}$, from $i = 2 \sim n-1$, excluding *Head* $(i = 1)$ and *Tail* $(i = n)$.

This research now defines a shape number (SN), derived from each of the rolling segments as

$$S_{\text{seq}} = \left\{ s_{\text{seg }2} s_{\text{seg }3} s_{\text{seg }4} \cdots s_{\text{seg }n-3} s_{\text{seg }n-2} s_{\text{seg }n-1} \right\}. \quad (2)$$

In short, $S = \{s_{\text{seg }i}, i = 2 \sim n-1\}$ is a collection of the ordered cyclic difference from each segment's continuous 3 chain codes. Thus, for each segment, SN is composed of three integer numbers as $S_{\text{seg }i} = \{s_{i-1}, s_i, s_{i+1}\}$, defined as

$$s_{\text{seg}_i} = \left\{ \begin{array}{c} s_{i-1} = c_{i-1} - c_{i+1}, \\ s_i = c_i - c_{i-1}, \\ s_{i+1} = c_{i+1} - c_i, \end{array} \right\}. \quad (3)$$

Since all CC is comprised of integers from $0 \sim 7$, SNs (s_i) are also retained between $0 \sim 7$. Thus, whenever s_i is derived as negative, we automatically take 8's compliment to correct it, with $s_i = s_i + 8$ if $s_i < 0$.

One such example of SN derived is illustrated in the lower part of Figure 11, where the SN is calculated from both directions of the chain code inside each segment: the *Start to End Chain Code* (SECC) as well as the *End to Start Chain Code* (ESCC). It is obvious that SECC and ESCC are different; therefore, the resulting *Start to End Shape Number* (SESN) and *End to Start Shape Number* (ESSN) are also derived different, albeit representing the exact same segment.

To ensure exclusive identification on the same 4-pixel segment, for both bidirectional CC and SN coding, in addition to all the same shape mirroring and rotational configuration, a simple unique identifier (ID) number is needed to match the rolling 4-pixel segments to the $k_1 \sim k_{12}$ shapes.

2.2.2. Unique Identifier (ID) Matching. In order to deal with such bidirectional, mirroring, and rotational segment ambiguity, the following rule has been applied to ensure a single SN identifier (ID) to match explicitly one $k_1 \sim k_{12}$

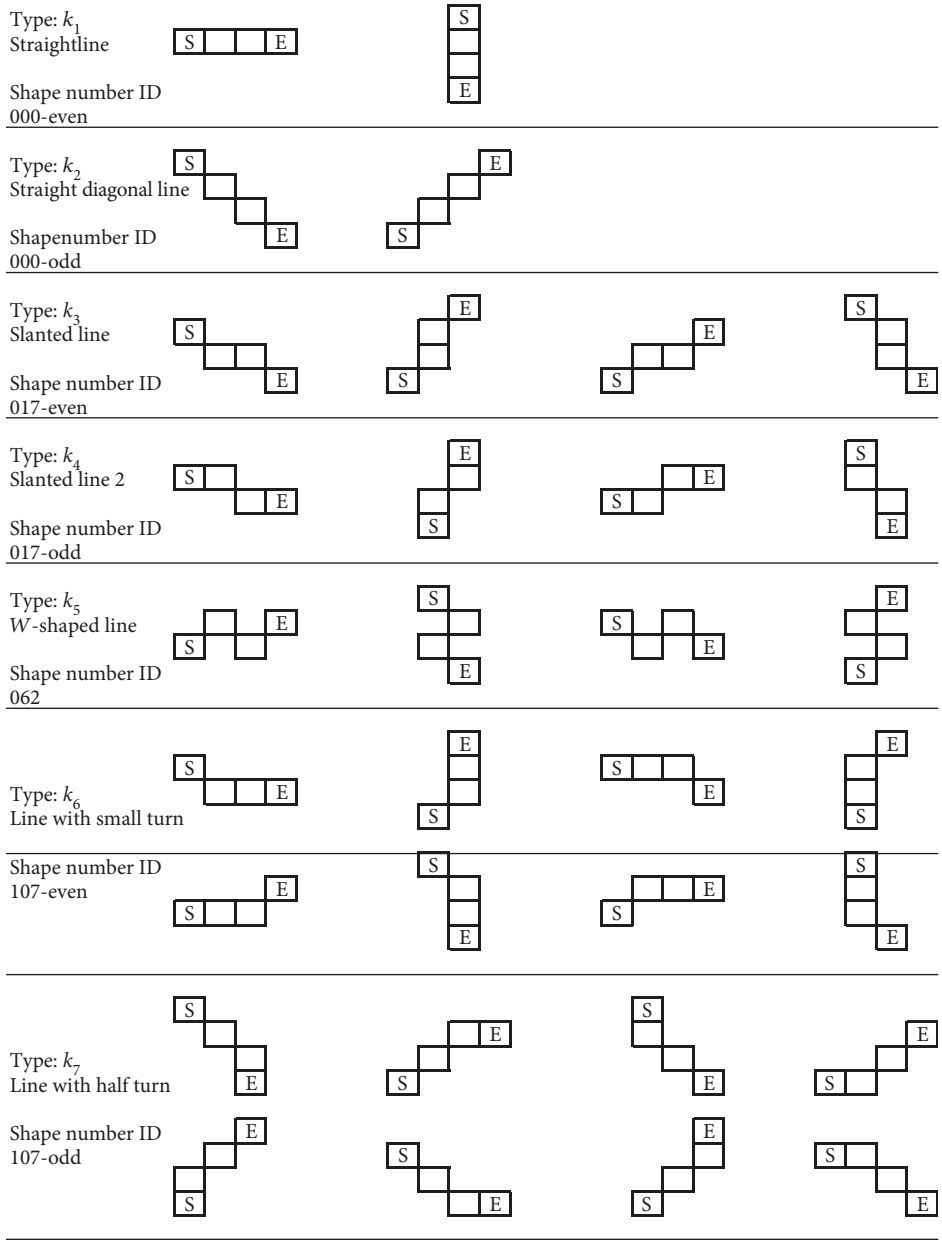


Figure 6: Four-pixel segments of the shapes for $k_1 \sim k_7$, with their associated rotation and mirror alternatives.

shape, for any given random shape number in the lengthy l_c contour skeleton backbone.

This is provided by examining all the SESN and ESSN for all the $k_1 \sim k_{12}$ shapes and capturing the combinatory relative adjacent arrangements from the 4-pixel segment geometry, i.e., reordering the representative numerals of SN to allow for the direct/diagonal connections, to make representation of the given shape.

Since the rolling window segmental SN $s_{seg\ i}$ will fall into the recognizable $k_1 \sim k_{12}$ shapes, the ID number can be derived from the known segment numbers as specified in Figure 9. Thus, the identifier is a unique number for each of the shape k_j, $j = 1 \sim 12$, such that when the rolling window covers a 4-pixel segment, by performing this numeral operation (algorithm), one will find the identifier.

The following Algorithm (1) outlines the unique identifier (ID)'s reorder methodology for all of the $k_1 \sim k_{12}$ shapes.

The rules for the identifier are stated as follows:

Unique—there exists one unique ID number for each of k_5, k_8, k_9, and k_{12} shapes.

Common—there exists *common* ID numbers for the following pairs: $\{k_1, k_2\}$, $\{k_3, k_4\}$, $\{k_6, k_7\}$, and $\{k_{10}, k_{11}\}$.

Distinguish—the aforementioned sets are discerned by the connection type of the center pixels, {direct or diagonal}, by checking its original CC number c_i, with even/odd numbers representing direct/diagonal, respectively.

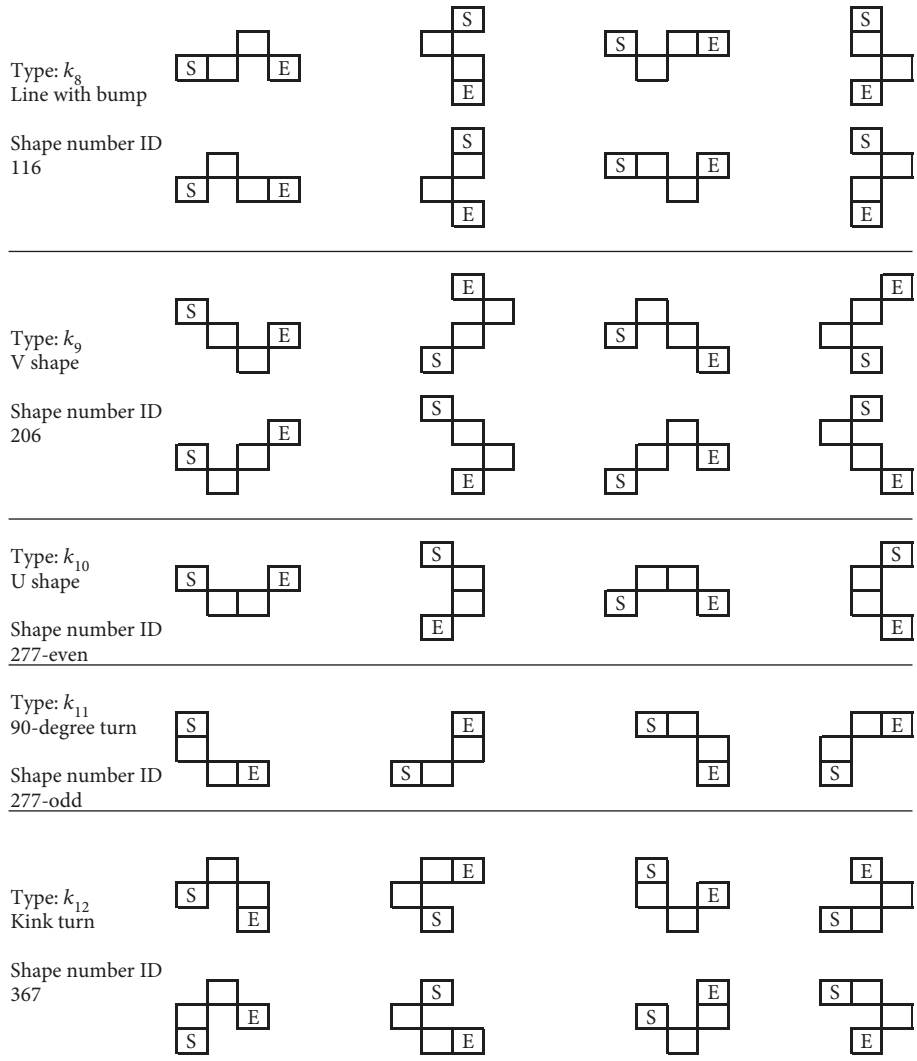


Figure 7: Four-pixel segments of the shapes for $k_8 \sim k_{12}$, with their associated rotation and mirror alternatives.

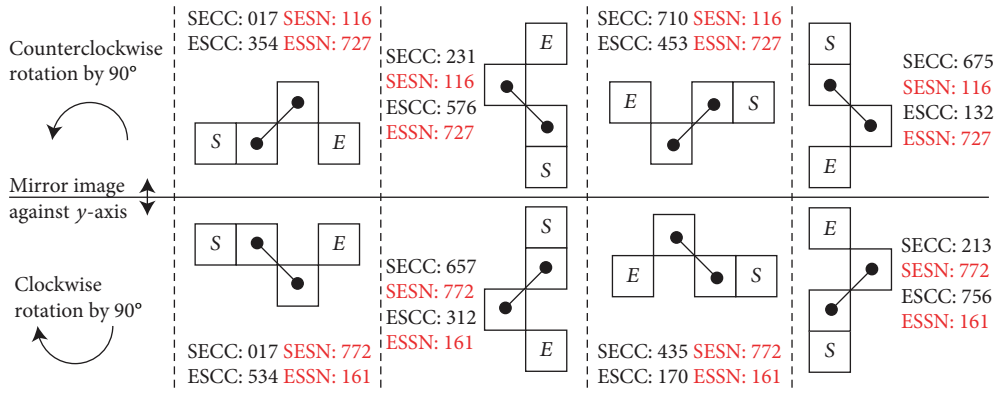


Figure 8: Rotational and mirror representation of k_8 shape 4-pixel segment.

Take for example the k_8SN, as there exist four different SN combinations: 116, 772, 727, and 161 form shaping of the eight different segments, as shown in the last row of Figure 8, due to coding of all the different mirrors/rotations and bidirectional CC.

After performing the abovementioned ID algorithm, we are able to uniquely transform all SN to the same identifier (ID) number: 116, as shown from Table 2.

Finally, the algorithm arrives with k_5, k_8, k_9, and k_{12} matching up with their respective ID numbers: 026, 116,

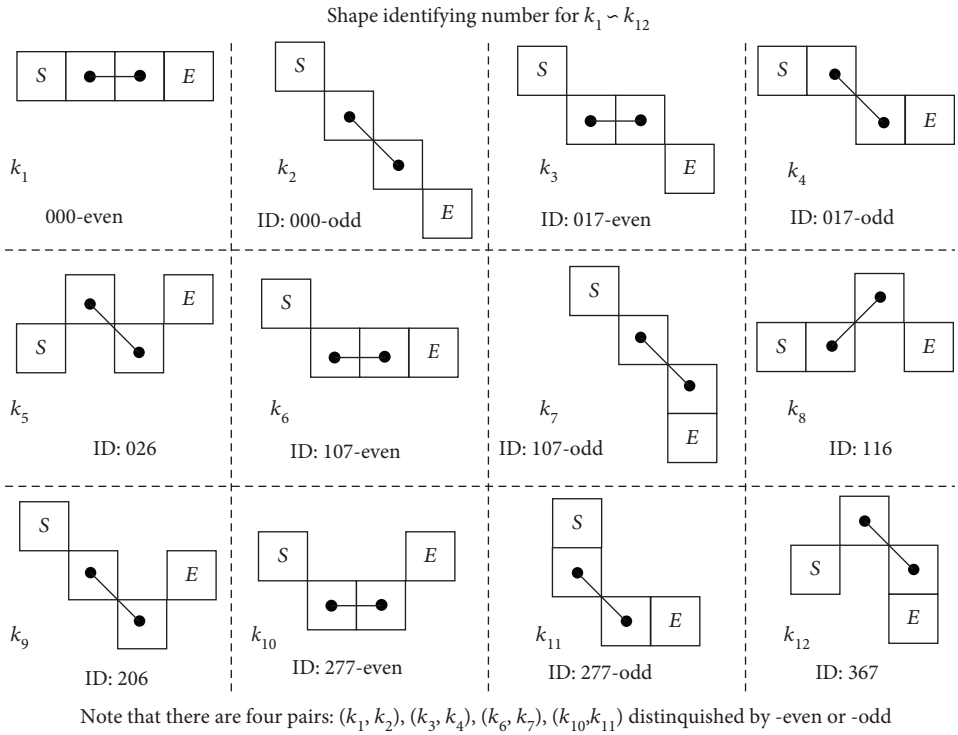


Figure 9: Geometric connectivity of the twelve independent $k_1 \sim k_{12}$ shapes for 4-pixel segments. The inner two direct connections (horizontal/vertical and diagonal) are connected with two additional outward pixels, accounting for the local 2D morphology change.

$$
\begin{aligned}
\mathrm{seg}_1 &= \boxed{c_1}, \\
\mathrm{seg}_2 &= c_1 \boxed{c_2} c_3, \\
\mathrm{seg}_3 &= c_2 \boxed{c_3} c_4, \\
&\vdots = \vdots \\
\mathrm{seg}_i &= c_{i+1} \boxed{c_i} c_{i+1}, \\
&\vdots = \vdots \\
\mathrm{seg}_{n-2} &= c_{n-3} \boxed{c_{n-2}} c_{n-1} \\
\mathrm{seg}_{n-1} &= c_{n-2} \boxed{c_{n-1}} c_n, \\
\mathrm{seg}_n &= \boxed{c_n}.
\end{aligned}
$$

Figure 10

206, and 367. In addition, the algorithm matches the pairs $\{k_1, k_2\}$, $\{k_3, k_4\}$, $\{k_6, k_7\}$, and $\{k_{10}, k_{11}\}$ commonly to 000, 017, 107, and 277, respectively. In order to distinguish between the pairs, the original {direct, diagonal} connection is once again used: by checking if the original c_i is either *even* or *odd*, then it can be trivially matched to the correct shape in the pair Table 3.

All the ID numbers are listed for $k_1 \sim k_{12}$ shapes here in Figure 9, derived from all the 64 shapes in Figures 6 and 7. Note that the common ID numbers are annotated with (-even/-odd) for distinguishing.

2.3. k_j Parameter Equation Representation. Now that the unique ID number is obtained, it is then ready to amass a collection of the different samples of a given l_c length, in order to retrieve the correction parameter for the different k_js, provided with the same DNA characteristics, i.e., with a fixed $l_p = 50$ nm.

2.3.1. Length Calculation with Coefficients. This is first done by identification on one individual DNA sample's contour, by summing up each shape component k_j's occurrence contribution for its segment's connection length. In other words, one identifies along the skeleton backbone and tallies the individual occurrences of the twelve k_js, multiplied by the corresponding connection length (1 or $\sqrt{2}$ pixel length) along with its correction coefficient. This makes the sum of all the length contributions equal to the contour length l_c as

$$l_c = r\left[\sum_{j=1}^{12} n_{kj} k_{jl} k_j + (l_{\mathrm{H}} + l_{\mathrm{T}})\right], \tag{4}$$

where $n_{kj}, j = 1 \sim 12$ is the number of occurrences of the type k_j shapes, provided from the identification along the skeleton backbone. k_j is the correction coefficient, and l_{kj} is the connection length (either 1 or $\sqrt{2}$ according to the k_j shape). l_{H} and l_{T} are the head and tail length, respectively, and finally r is the AFM image pixel resolution.

From Figure 9, $l_{k_1} \sim l_{k_{12}}$ length is ordered in Table 4.

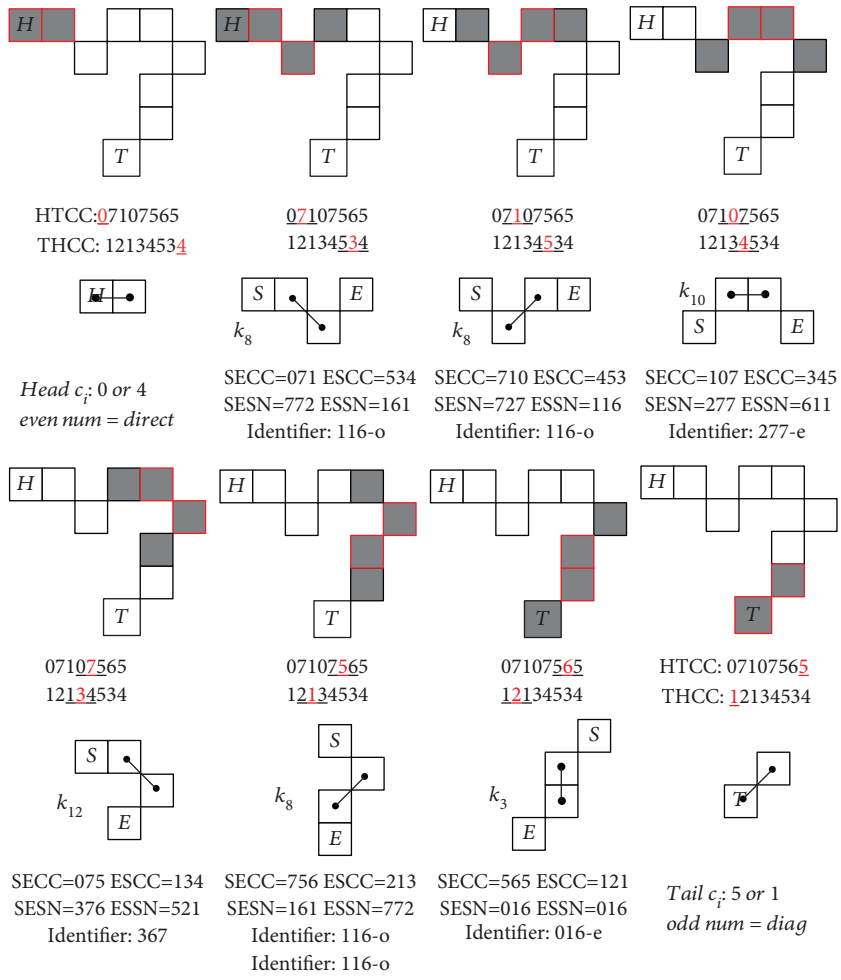


Figure 11: Consecutive determination of the 4-pixel segment shape number through the rolling window. Two sets of chain code provided HTCC (head to tail) and THCC (tail to head), with their associated local segment CC, also bidirectional: SECC (start to end) and ESCC (end to start), which leads to their derived SESN (start to end) and ESSN (end to start).

Require: k_j shapes, $j = 1 \sim 12$ and their respective SESN or ESSN $s_{kj} = s_1 s_2 s_3$
Ensure: Unique ID numbers for all $k_1 \sim k_{12}$ shapes.
1: **for** $j = 1 \sim 12$ **do**
2: take the k_j's SN: $s = \{s_1, s_2, s_3\}$
3: **if** $s_1 \geq 4$ **then**
4: take 8's complement for this sequence of s, for bi-direction and mirroring re-ordering, but
5: **if** $s_2 = 0$ or $s_3 = 0$ **then**
6: keep s_2 or s_3 zero
7: **else**
8: $s_2 = 8 - s_2$ and $s_3 = 8 - s_3$
9: end if
10: end if
11: Take the new S (with $s_1 \leq 4$), and further re-order s_2 and s_3, to match the rotational varieties, by
12: **if** $s_2 > s_3$ **then**
13: switch s_2 with s_3
14: **else**
15: retain original order
16: **end if**
17: Find final ID number for given SN s_{kj}
18: end for

Algorithm 1 : ID number derived from shape number.

Table 2: Example ID number retrieval for k_8 shape numbers.

	Original k_8 SN	116	772	727	161
Step one	Check s_1, take 8's complement if $s_1 > 4$	116	$\underbrace{116}_{8's\,\text{comp}}$	$\underbrace{161}_{8's\,\text{comp}}$	161
Step two	Reorder s_2 and s_3 digits when $s_2 > s_3$	116	116	$1\ \underbrace{16}_{\text{reorder}}$	$1\ \underbrace{16}_{\text{reorder}}$
	Final unique k_8 ID			**116**	

Table 3: ID number derivation from SN for all $k_1 \sim k_{12}$ shapes.

Shape	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9	k_{10}	k_{11}	k_{12}
ID number	**000even**	**000odd**	**017even**	**017odd**	**026**	**107even**	**107odd**	**116**	**206**	**277even**	**277odd**	**367**
Original SN	000	000	017	017	026	107	107	116	206	277	277	367
			071	071	062	170	170	161	260	611	611	376
						701	701	727	602			512
						710	710	772	620			521

Table 4: Connection length for $l_{k_1} \sim l_{k_{12}}$.

Shape	k_1	k_2	k_3	k_4	k_5	k_6
Length	1	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{2}$	1
Shape	k_7	k_8	k_9	k_{10}	k_{11}	k_{12}
Length	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{2}$

2.3.2. Matrix Form for Inverse Calculation. The second step is to collect a sufficient amount of representation of this same type of biopolymer samples and list all the length equations based on these samples. The logic is that with multiple samples of the same kind, imaged under the same pixel resolution, the k_j shapes collectively represent the same type of twist/turn, resulting in the same length contribution for the same class of biopolymers l_c.

Combining equation (4) with the associated l_{kj} values in Table 4, we arrive at

$$\begin{aligned} lm, c = r\big[&\big(n_{m,k_1}k_1(1) + n_{m,k_2}k_2(\sqrt{2}) + n_{m,k_3}k_3(1) \\ &+ n_{m,k_4}k_4(\sqrt{2}) + n_{m,k_5}k_5(\sqrt{2}) + n_{m,k_6}k_6(1) \\ &+ n_{m,k_7}k_7(\sqrt{2}) + n_{m,k_8}k_8(\sqrt{2}) + n_{m,k_9}k_9(\sqrt{2}) \\ &+ n_{m,k_{10}}k_{10}(1) + n_{m,k_{11}}k_{11}(\sqrt{2}) + n_{m,k_{12}}k_{12}(\sqrt{2})\big) \\ &+ (l_{m,\mathrm{H}} + l_{m,\mathrm{T}})\big], \end{aligned} \tag{5}$$

for any given backbone skeleton length $l_{m,c}$, given the index m'th sample. Equation (5) can be represented with a matrix form, with

$$r(NDK + B) = L, \tag{6}$$

where

$$N = \begin{bmatrix} n_{1,k_1} & n_{2,k_2} & \cdots & n_{1,k_{12}} \\ n_{2,k_1} & n_{2,k_2} & \cdots & n_{2,k_{12}} \\ \vdots & \vdots & \ddots & \vdots \\ n_{m,k_1} & n_{m,k_2} & \cdots & n_{m,k_{12}} \end{bmatrix},$$
$$K = \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_{12} \end{bmatrix}, \tag{7}$$

and

$$B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, \tag{8}$$

and

$$L = \begin{bmatrix} l_{c,1} \\ l_{c,2} \\ \vdots \\ l_{c,m} \end{bmatrix} \tag{9}$$

provided $D = \mathrm{diag}\,(1, \sqrt{2}, 1, \sqrt{2}, \sqrt{2}, 1, \sqrt{2}, \sqrt{2}, \sqrt{2}, 1, \sqrt{2}, \sqrt{2})$ is a 12-by-12 square matrix, such that $\mathrm{DK} = [k_1, \sqrt{2}k_2, k_3, \sqrt{2}k_4, \sqrt{2}k_5, k_6, \sqrt{2}k_7, \sqrt{2}k_8, \sqrt{2}k_9, k_{10}, \sqrt{2}k_{11}, \sqrt{2}k_{12}]^{\mathrm{T}}$.

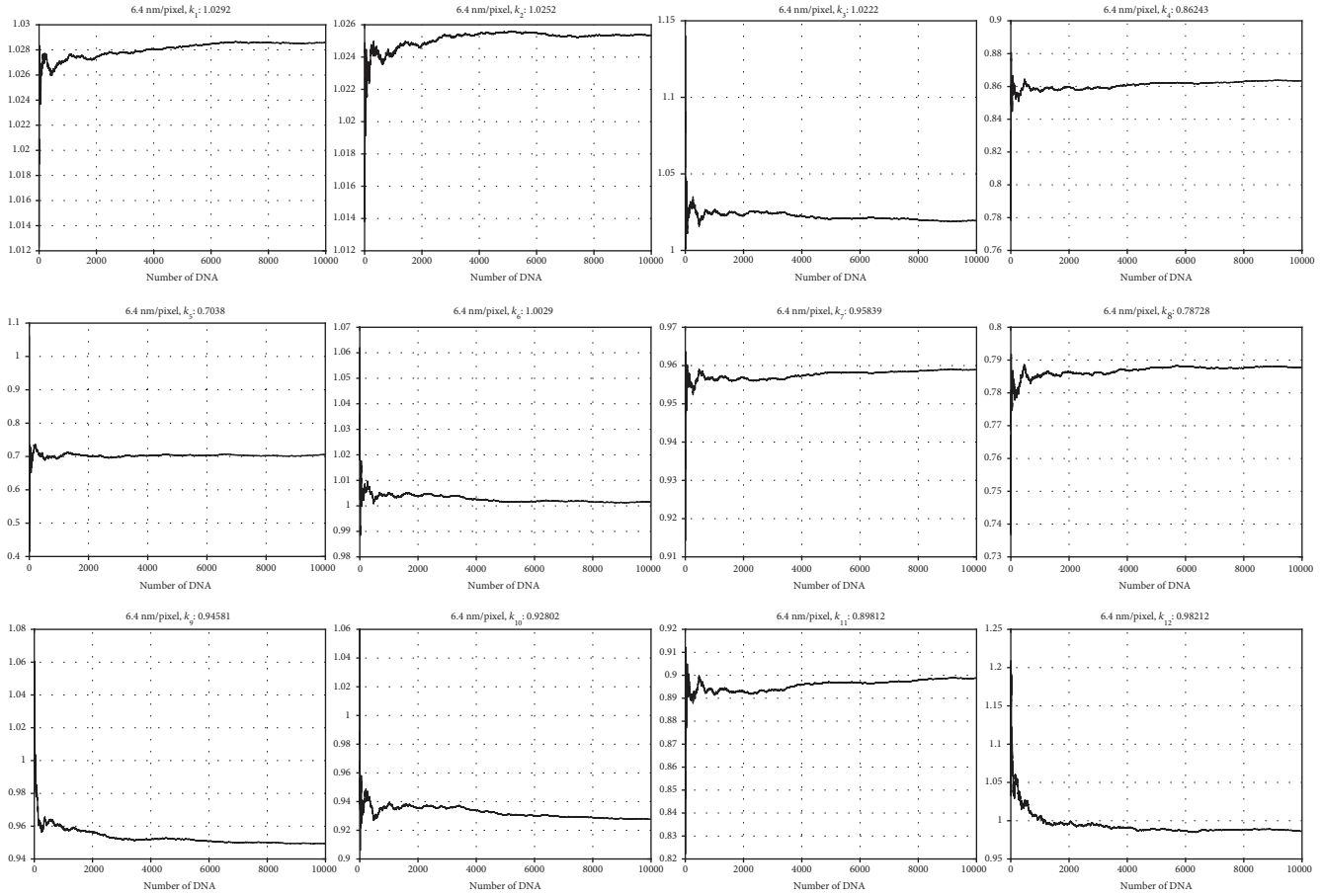


FIGURE 12: Convergence of all $k_1 \sim k_{12}$ coefficients, with growing number of $m = 2.1 \times 10^6$ image samples. Resolution at 6.4 nm/pixel, midrange from all simulation data.

Note that $b_m = l_{h,m} + l_{t,m}$ is the *head* and *tail* (boundary) connection length summation. Also note that N is an m-by-12 matrix, K is 12 by 1, and B and L are both m-by-1 matrices.

The final procedure here is to derive matrix K using a standard linear regression and find the best fit for the $k_1 \sim k_{12}$ value. The final results are presented in the next section.

3. Contour Parameter Calibration Result

In order to guarantee convergences of the $k_1 \sim k_{12}$ coefficients, different known values of l_c and r single-pixel-width AFM images were simulated for k_j calibration. Due to the combinations of different l_c and r, plus a surplus amount of samples for each (l_c, r) pair, a total of 58,800,000 images were generated.

All the simulated images are based on DNA characteristics, as mentioned in Introduction, where all the samples have the same persistance length of $l_p = 50$ nm.

The different lengths calculated ranged from 340 to 1020 nm, for every 34 nm, and the different resolution r is simulated between 5.1 and 7.8 nm/pixel, with a 0.1 nm interval. Thus, there are 21 different l_c scales, along with 28 altering r, making a total of $21 \times 28 = 588$ test cases. Each case is studied with 10,000 DNA images, for sufficient representation on k_js. In order words, the test index $m = 10{,}000$ was used for equation (6).

3.1. Convergence of the k_j Coefficient. This research first checks the convergence of all k_j coefficients, given a growing number of image files, i.e., growing number of m in equation (6). The results are illustrated in Figure 12.

All coefficients verify its convergence when given more than 0.5 million samples and remain constant with fluctuation of less than 0.01% after 1 million samples. This result is verified for all resolutions $r = 5.1 \sim 7.6$ nm/pixel, showing similar trend for all k_js.

3.2. Linear Variation of k_j Dependence on Resolution r. With the convergence for all the k_j coefficients confirmed for all different r, the relationship for each $k_j(\cdot)$ as a function of r, i.e., the linear fit for $k_j(r)$ results, is found in Figure 13.

It is clear that using the converged k_j values for a specified r, the following linear fit equation,

$$k(r) = mr + b, \tag{10}$$

results with a table that contributes to all the twelve different coefficients; it is provided in Table 5.

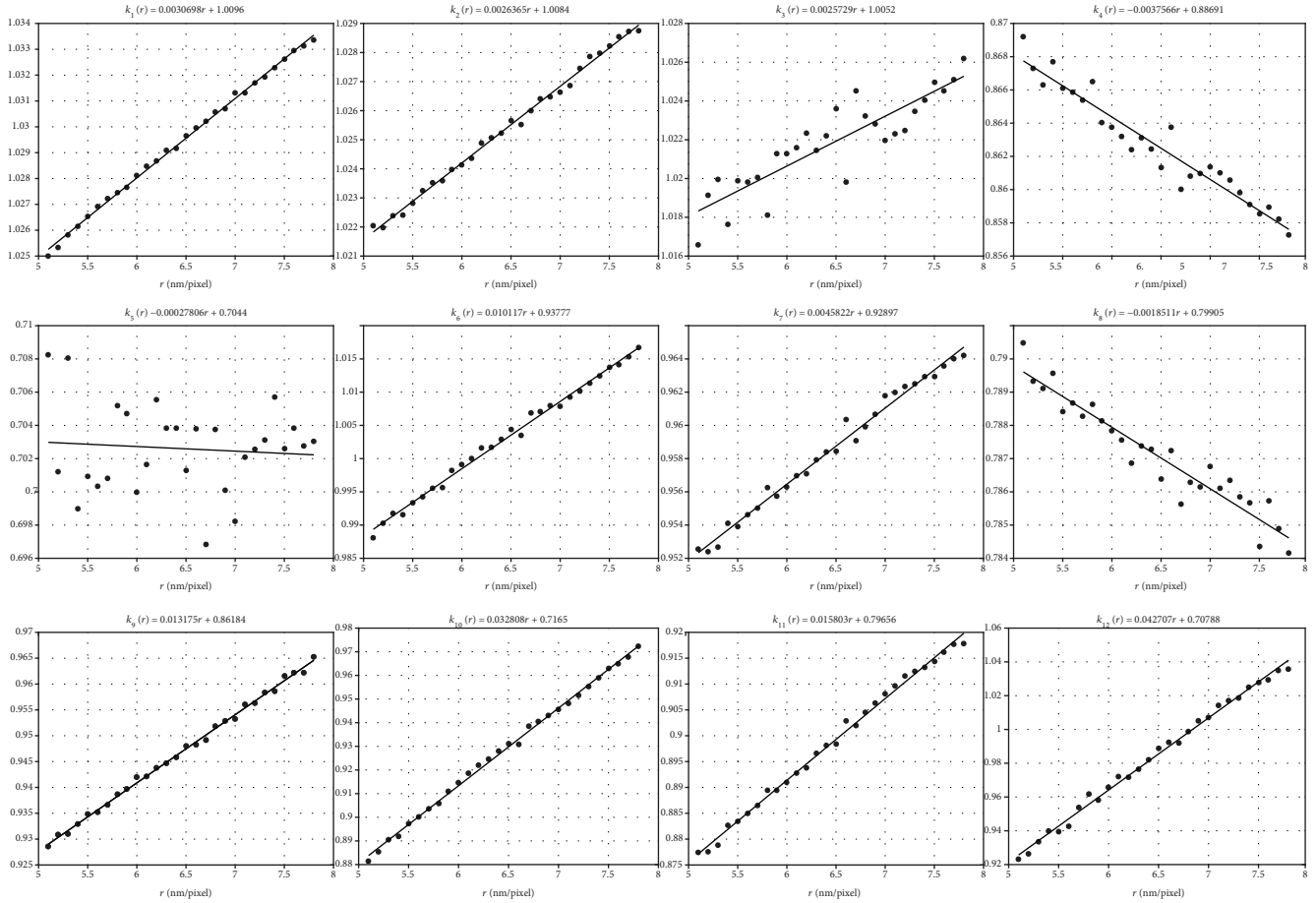


Figure 13: Linear fit of all $k_1 \sim k_{12}$ coefficients, with respect to resolution r, i.e., $k_j(r) = mr + b$, as in equation (10).

Table 5: Coefficient table of m and b in $k_j(r) = mr + b$.

$k(r)$	m	b
k_1	0.0030698	1.0096
k_2	0.0026365	1.0084
k_3	0.0025729	1.0052
k_4	-0.0037566	0.88691
k_5	-0.00027806	0.7044
k_6	0.010117	0.93777
k_7	0.0045822	0.92897
k_8	-0.0018511	0.79905
k_9	0.013175	0.86184
k_{10}	0.032808	0.7165
k_{11}	0.015803	0.79656
k_{12}	0.042707	0.70788

3.3. Performance with Shape Modification Coefficient. The above sections result in the calibration correction in equation (10) and can be used for the unknown l_c estimation. In order to demonstrate such performance, the coefficient in equation (10) is used for the shape estimator L_S and compared alongside the DNA estimator L_{DNA} and the Freeman estimator L_F.

All estimators L_S, L_{DNA}, and L_F are compared with different length l_c and resolution r. All the estimators are applied for the same simulated pixel images and compared against the readily known l_c for error calculation.

Tables 6, 7, and 8 outline the calculation error for different r settings. It shows that the L_S estimator has an averaged relative error maxed at 0.07%, performing with an order of magnitude difference from the L_{DNA} estimator, and two orders of magnitude smaller than L_F. The relative error translates to an absolute value of maximum 0.20 nm for the $r = 5.1$ nm/pixel, well below the resolution, making L_S ideal for l_c estimation.

Since the error is averaged amongst the 100,000 samples provided, its standard deviation (STD) in nm is also an indicator for quantitative analysis. The L_S estimator also has a smaller standard deviation compared to both L_{DNA} and L_F, against a growing l_c contour estimated.

4. Conclusion and Future Direction

This research provided a novel way to estimate digitized contour length, in a general way that is applicable towards all kinds of contour curvature. Utilizing a localized shape

Table 6: Error analysis of L_S, L_{DNA}, and L_F, at $r = 5.1$ nm/pixel.

$r = 5.1$ nm/pixel		Error		
Estimator	$l_c(nm)$	Relative (%)	Absolute (nm)	std (nm)
L_S	340	0.05	0.19	3.6
	680	0.00001	0.00009	4.51
	1020	0.02	0.20	5.23
L_{DNA}	340	0.26	0.99	4.71
	680	0.24	1.67	6.18
	1020	0.24	2.45	7.36
L_F	340	3.42	11.63	4.86
	680	3.40	23.12	6.38
	1020	3.39	34.63	7.60

Table 7: Error analysis of L_S, L_{DNA}, and L_F, at $r = 6.4$ nm/pixel.

$r = 6.4$ nm/pixel		Error		
Estimator	l_c(nm)	Relative (%)	Absolute (nm)	std (nm)
L_S	340	0.06	0.23	4.77
	680	0.0005	0.003	5.92
	1020	0.02	0.24	6.92
L_{DNA}	340	0.35	1.21	5.80
	680	0.33	2.28	7.57
	1020	0.32	3.31	9.06
L_F	340	2.87	9.78	5.94
	680	2.85	19.43	7.76
	1020	2.84	29.03	9.29

Table 8: Error analysis of L_S, L_{DNA}, and L_F, at $r = 7.7$ nm/pixel.

$r = 7.7$ nm/pixel		Error		
Estimator	l_c (nm)	Relative (%)	Absolute (nm)	std (nm)
L_S	340	0.07	0.24	5.93
	680	0.004	0.03	7.49
	1020	0.02	0.29	8.76
L_{DNA}	340	0.43	1.47	6.93
	680	0.40	2.73	9.11
	1020	0.39	4.01	10.86
L_F	340	2.32	7.91	7.06
	680	2.29	15.61	9.28
	1020	2.28	23.33	11.07

connection approach, and correct upon the local connectivity between pixels, this algorithm accounts for both resolution and the sample stiffness.

This research is general in the local 4-pixel segment identification method and extensible towards extension to more pixel elements. The general idea stands that a single-width pixel contour's digital shape recognition is applicable towards all images acquired from different systems, not only with the AFM family but also optical microscopy systems, electron microscopy systems, and many others.

Experimental verification is also needed for future research, provided with calibrated accurate sample length l_c from DNA samples or other biopolymer samples imaged with AFM systems.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The authors would like to thank the funding provided from the Ministry of Science and Technology, Taiwan. The research is supported through grant number MOST 105-2221-E-011-056.

References

[1] G. Binnig, C. F. Quate, and C. Gerber, "Atomic force microscope," *Physical Review Letters*, vol. 56, no. 9, pp. 930–933, 1986.

[2] D. Y. Abramovitch, S. B. Andersson, L. Y. Pao, and G. Schitter, "A tutorial on the mechanisms, dynamics, and control of atomic force microscopes," in *2007 American Control Conference*, pp. 3488–3502, New York, NY, USA, July 2007.

[3] K. D. Dorfman, "The statistical segment length of dna: opportunities for biomechanical modeling in polymer physics and next-generation genomics," *Journal of Biomechanical Engineering*, vol. 140, no. 2, article 020801, 2018.

[4] D. Cramb, Z. Leonenko, D. Merkle, and S. Lees-Miller, "Atomic force microscopy at the surface of chemistry and biology," *Recent Research Developments in Physical Chemistry*, Transworld Research Network, 2002.

[5] H. Wang and J. N. Milstein, "Simulation assisted analysis of the intrinsic stiffness for short DNA molecules imaged with scanning atomic force microscopy," *PLoS One*, vol. 10, no. 11, article e0142277, 2015.

[6] A. K. Mazur and M. Maaloum, "Atomic force microscopy study of DNA flexibility on short length scales: smooth bending versus kinking," *Nucleic Acids Research*, vol. 42, no. 22, pp. 14006–14012, 2014.

[7] H. Freeman, "On the encoding of arbitrary geometric configurations," *IEEE Transactions on Electronic Computers*, vol. EC-10, no. 2, pp. 260–268, 1961.

[8] T. S. Spisz, Y. Fang, R. H. Reeves, C. K. Seymour, I. N. Bankman, and J. H. Hoh, "Automated sizing of DNA fragments in atomic force microscope images," *Medical and Biological Engineering and Computing*, vol. 36, no. 6, pp. 667–672, 1998.

[9] C. Rivetti and S. Codeluppi, "Accurate length determination of DNA molecules visualized by atomic force microscopy: evidence for a partial b-to a-form transition on mica," *Ultramicroscopy*, vol. 87, no. 1-2, pp. 55–66, 2001.

[10] A. Sanchez-Sevilla, J. Thimonier, M. Marilley, J. Rocca-Serra, and J. Barbet, "Accuracy of AFM measurements of the contour length of DNA fragments adsorbed on mica in air and in aqueous buffer," *Ultramicroscopy*, vol. 92, no. 3-4, pp. 151–158, 2002.

[11] E. Ficarra, L. Benini, E. Macii, and G. Zuccheri, "Automated DNA fragments recognition and sizing through AFM image processing," *IEEE Transactions on Information Technology in Biomedicine*, vol. 9, no. 4, pp. 508–517, 2005.

[12] C. Rivetti, "A simple and optimized length estimator for digitized dna contours," *Cytometry Part A*, vol. 75A, no. 10, pp. 854–861, 2009.

[13] A. Sundstrom, S. Cirrone, S. Paxia et al., "Image analysis and length estimation of biomolecules using AFM," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 6, pp. 1200–1207, 2012.

[14] L. Lam, S. W. Lee, and C. Y. Suen, "Thinning methodologies-a comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, pp. 869–885, 1992.

[15] M.-C. Hsiao, "DNA contour length estiamtor for shape number from AFM image," Master's thesis, National Taiwan University of Science and Technology, 2016.