Un-Wrapping Multi-Passed images

CS 229 Final Project
Stewart Koppell

Abstract

The ability to image individual organic molecules with atomic resolution would be a major breakthrough for biology and pharmacy. Multi-Pass Transmission Electron Microscopy (MPTEM) has recently been proposed to accomplish this goal by reducing damage an order of magnitude below current state of the art techniques [1]. One of the main limits of MPTEM is phase wrapping. If the effects of phase wrapping can be removed through post-processing, MPTEM could improve the state of the art by a second order of magnitude.

1 Introduction

When imaging a delicate microscopic structure, the resolution-limiting factor is often the dose tolerance of the sample rather than the quality of the microscope optics. A transmission electron microscope (TEM) can create atomic-resolution images of sturdy samples like metallic nanoparticles, but cannot image biological molecules at the same resolution before their structure is dramatically altered by the imaging process. By passing each probe particle (electrons, for a TEM) m times through the sample (and using m times fewer particles), it's possible to decrease the damage done to the by a factor of m at constant signal-to-noise (see the referenced paper[1] for an explanation of this counter-intuitive effect).

In a multi-pass transmission electron microscope (MPTEM), electrons are passed through a thin sample and store information about it in their phase. Phase is an angular property measured in radians, so $\phi = \phi + 2\pi$. If a particular region of the sample causes electron phases shifts of ϕ, then the electron will have a phase of $m\phi$ after m passes. When $m\phi$ becomes large, it may not be clear how many multiples of 2π the phase has shifted. This problem is called phase wrapping. It is the goal of this project to "un-wrap" multi-passed images. Figure 1 shows the phase of an electron passing through a protein (HIV-1 Gag) m=1, 8, 16, and 32 times.

Given a multi-passed image $\hat{I} \in \mathbb{R}^{n \times n}$ and integer m, we want to reconstruct I such that

$$mI = \hat{I} + 2\pi N$$

The matrix N is the wrapping number: the number of times each pixel's phase has wrapped (each element of N should be less than m). With no extra information, the number of solutions is enormous: m^{n^2}. However, it seems reasonable to expect that neighboring pixels should be correlated with each other. Proteins, for example, are large collections of a small variety of atoms with a small variety of bond lengths. And at a slightly larger scale, groups of atoms are arranged into 1 of 21 amino acids. If the value of any particular pixel can be reliably predicted -to within 2π - by the values of pixels in it's vicinity, then we can un-wrap

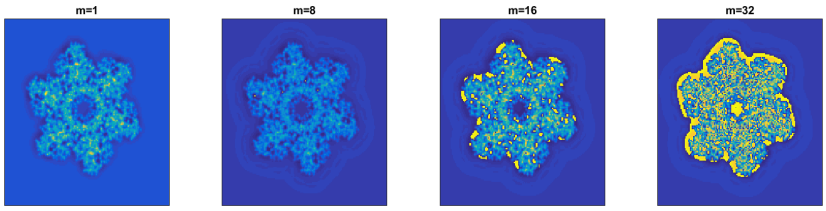


Figure 1: Phase wrapping for the protein HIV-1 Gag.

the image by solving a linear system of equations for N. This will be described in more detail in the next sections.

2 Comparison to Related Work

Phase unwrapping algorithms are well-studied and specialized to a wide variety of sub-problems[3]. In phase microscopy, these algorithms are used to correct mild cases of phase wrapping, but at some point the problem is considered intractable (this is known as the "interpretability problem" to microscopists). In the context of multi-passing, we have slightly more information than in the general case. We are mostly interested in thin samples, which don't phase wrap for $m = 1$ (they are "weak phase objects"). So when multipassing, our image is a superposition of m identical weak phase objects.

3 Method

Two methods will be presented in this paper: k-means and EM.

3.1 K-Means

Suppose, through examining training data, we obtain a library of characteristic $r \times r$ regions $\{\mathcal{L}_k\}$ so that we can approximate any given un-wrapped image I by stitching together copies of these regions. Then we should be able to approximate $\hat{I}$ by stitching together elements from $\{\text{mod}(m \times \mathcal{L}_k, 2\pi)\}$. We'll use k-means to create the library. In order to decrease the number of degrees of freedom the library must represent, we can cluster regions according to the amplitude of their Fourier transform, which is translation invariant. In addition, we can set the mean of each characteristic region to 0 so that the same library can be used in proteins of different thickness. Then, to un-wrap an image $\hat{I}$:

1. For each $r \times r$ region $\hat{R}$ in $\hat{I}$: for each image $\mathcal{L}$ in the library, find the 2-dimensional integer translation (t_x, t_y) which minimizes the loss
   $J = \sum_{i,j}(\hat{R}_{i,j} - \text{mod}(m \times \mathcal{L}_{i+t_x, j+t_y}, 2\pi))^2$.

2. Associate each $\hat{R}$ with the image from the library which has the smallest optimal loss.

3. Record the number of times each pixel in the library image wraps, $N \in \{0, 1, ..., m\}^{\otimes r \times r}$

4. The unwrapped region is $R = (\hat{R} + 2\pi N)/m$ up to a uniform integer offset.

Once we have the set of unwrapped regions $\{R\}$, we need to find the offset C_R for each region. Then we can stitch the regions into I. Finding $\{C_R\}$ is much easier if the regions overlap. Also, we can increase the confidence of our predictions by setting a pixels wrapping number according to the prediction of multiple un-wrapped regions. Assuming we've chosen $\{R\}$ so they overlap, we can find the constant offsets by:

1. Set C_R=0 for the first region.
2. Looping over all regions in $\{R\}$, choose C_R to minimize the discrepancy in R_{ij} of the overlapping pixels of other regions for which N_R has already been set.
3. Iterate the previous step several times if necessary.
4. The entire image is now determined up to an irrelevant global offset.

3.2 EM

Let $x^{(i)} \in \mathbb{R}^{3\times 3}$ be 9 pixel regions of the wrapped image $\hat{I}$. We'll define latent variables $z^{(i)} \in Z$ (Z will be defined momentarily) so that region R_i in the un-wrapped image I is $x^{(i)} + z^{(i)}$. We'll model $z^{(i)}$ as a multinomial random variable: $p(z^{(i)} = N) = \phi_N$. One immediate problem is the size of Z: if each pixel might be wrapped m times, then naively, $|Z| = m^9$. It will be much more practical to work in a smaller subspace: let's only admit z such that the central pixel has wrapped 0 times (the region will then be defined up to a constant offset) and neighboring pixels may not be wrapped twice relative to each other. In this case, $|Z| = 1155$ for all m.

In order to implement the EM algorithm, we need to define $p(x^{(i)}|z^{(i)})$. To do this, we'll assume that the finite-element derivatives measured in the 9 pixel regions are distributed normally with mean μ and covariance Σ. Define

$$D = [d^{[1]}_{\uparrow}, d^{[1]}_{\nearrow}, d^{[1]}_{\rightarrow}, d^{[1]}_{\searrow}, d^{[1]}_{\downarrow}, d^{[1]}_{\swarrow}, d^{[1]}_{\leftarrow}, d^{[1]}_{\nwarrow}, d^{[2]}_{\updownarrow}, d^{[2]}_{\leftrightarrow}, d^{[2]}_{\nearrow\swarrow}, d^{[2]}_{\nwarrow\searrow}]$$

where $d^{[1]}_a$ is a first-order derivative in direction a, while $d^{[2]}_a$ is a second-order derivative spanning directions a. For example,

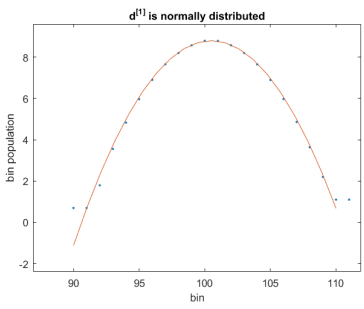
$$d^{[1]}_{\uparrow}(R) = R_{12} - R_{22}, \quad d^{[1]}_{\nearrow}(R) = R_{13} - R_{22}, \quad d^{[2]}_{\updownarrow}(R) = R_{12} - 2R_{22} + R_{13}$$

In practice, I found that opposite first-order derivatives (e.g. $d^{[1]}_{\uparrow}$ and $d^{[1]}_{\downarrow}$) were so highly correlated that the covariance matrix became singular. Therefore, I generally left out the first 4 elements of D. So $p(x^{(i)}|z^{(i)}) = p(D(x^{(i)} + z^{(i)})) \sim \mathcal{N}(\mu, \Sigma)$ with $\mu \in \mathbb{R}^8$ and $\Sigma \in \mathbb{R}^{8\times 8}$. With these definitions, the EM algorithm is

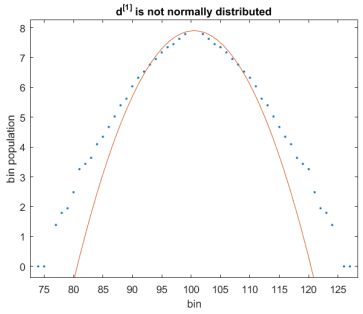
$$\text{(E-Step)} \quad Q_i(z^{(i)}) = p(x^{(i)}|z^{(i)})\phi_{z^{(i)}} / \sum_z p(x^{(i)}|z)\phi_z$$

$$\text{(M-Step)} \quad \phi_z = \sum_i Q_i(z)/m \qquad \mu = \sum_i \sum_z Q_i(z) D(x^{(i)} + z) / \sum_i \sum_z Q_i(z)$$

$$\Sigma = \sum_i \sum_z Q_i(z)(D(x^{(i)} - z))(D(x^{(i)} - z))^T / \sum_i \sum_z Q_i(z)$$



(a) Log bin populations of Perlin noise derivatives, quadratic fit



(b) Log bin populations of HIV-1 Gag derivatives, quadratic fit

Figure 2: First order derivatives of Perlin noise are normally distributed. In general, this isn't true for proteins.

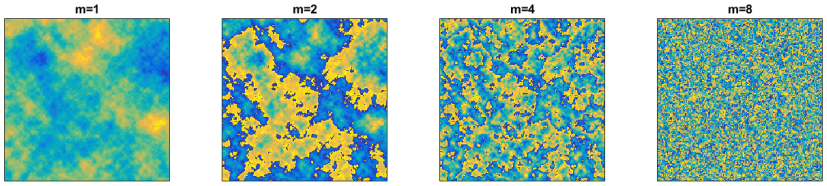


Figure 3: Perlin noise, wrapped m times.

4 Results

For the purpose of designing and testing these algorithms, I used Perlin noise rather than images of proteins. Perlin noise is structured, so it can plausibly be unwrapped. It is simple to quickly generate large Perlin Noise training sets. In addition, the first and second order derivatives of Perlin noise are normally distributed (see figure 2), which doesn't generally seem to be true for proteins. The accuracy of the algorithm can be described in two ways:

1. pixel accuracy: the fraction of pixels for which the wrapping number $N_{i,j}$ was correctly assigned.

2. block accuracy: the fraction of regions for which all pixels were assigned the correct wrapping number.

4.1 K-Means

Once an un-wrapped region R is calculated by finding the wrapping numbers N so that $R = (\hat{R} + 2\pi N)/m$, we can measure the accuracy of the process by comparing R to the original data. However N is only determined up to an integer constant. We can remove this degree of freedom by subtracting an integer C_N from N so that $N_{1,1} = 0$. Table 1 shows the pixel accuracy achieved using the k-means approach. The block accuracy was essentially 0 in every case.

m=4	r=3	r=4	r=5		m=8	r=3	r=4	r=5
k=30	0.49	0.44	0.37		k=30	0.42	0.31	0.26
k=50	0.49	0.45	0.35		k=50	0.36	0.25	0.20
k=70	0.49	0.45	0.51		k=70	0.42	0.30	0.31

Table 1: K-Means pixel accuracy for region size $r \times r$, using k k-means centroids, from Perlin noise wrapped m times. For $m = 4$, random guesses would be correct 32% of the time (since one pixel is always 0 and the others can take 1 of 4 values). For $m = 8$, random guesses would be correct 22% of the time.

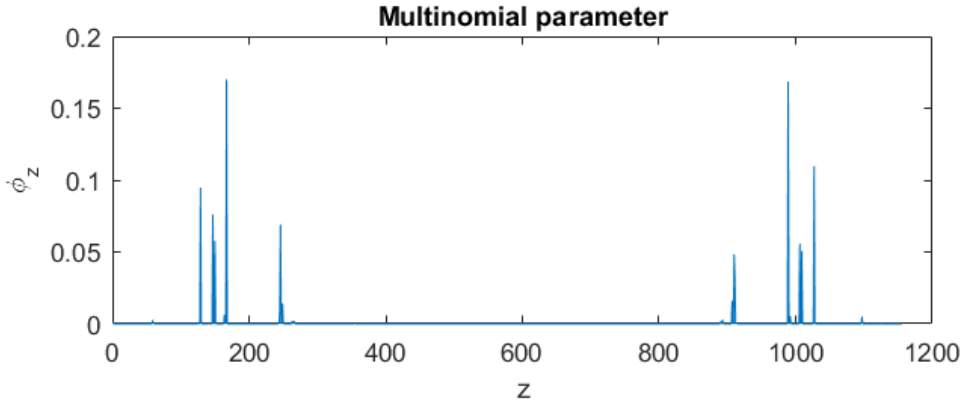


Figure 4: Distribution of multinomial parameter for z in EM algorithm

4.2 EM

The EM method for unwrapping thrice-wrapped (m=3) Perlin noise achieved 67% pixel accuracy and 46% block accuracy. Random guesses would have a block accuracy of 0.08%.

5 Conclusion and Future Work

While these techniques performed substantially better than random guessing, they were not consistent enough to unwrap multi-passed images. An important next step would be to implement a stitching algorithm to combine overlapping regions, which would help increase the pixel accuracy (by comparing results from different regions) and fix the relative offsets of each region. Figure 4 shows ϕ_z as calculated in the EM algorith. There is clearly a symmetry, possibly corresponding to a spacial inversion of the pixels ($z_{i,j} \rightarrow z_{j,i}$) which could be explicitly enforced to reduce the size of $|Z|$. The sparsity of ϕ_z suggest that the k-means approach, if carefully implemented, could have merit: only a few z are important in reconstructing the image.

6 References

[1] Juffmann, T. et. al., "Multi-Pass Transmission Electron Microscopy", Scientific Reports 7, Article number: 1699 (2017)
[2] Protein Data Bank website: https://www.rcsb.org/
[3] Baldi A., Bertolino F., Ginesu F. (2000) Phase Unwrapping Algorithms: A Comparison. In: Jacquot P., Fournier JM. (eds) Interferometry in Speckle Light. Springer, Berlin, Heidelberg