PHYS180 Project 7
Protein Folding

Rey Cervantes, Owen Morehead, Karoli Clever

May 28, 2021

1. Myosin Simulation Race

my.py Simulation

In the model used in my.py, we simulate a motor protein on a walk along a one-dimensional representation of an actin filament. The aim of this problem is to reduce the amount of time that it takes to complete a "race," where the protein travels 30 units on the actin filament. There are various parameters that can be varied in the code:

- The ComingOnRate/ComingOffRate dictates the process of ATP binding in the simulation. These parameters influence how the ends of the protein bind to actin sites. The ComingOnRate regulates the rate at which new ATP molecules come to a loose end of the protein in order to bind to another site. This should be set to be quick or else the protein will take a lot longer abiding freely in the medium instead of being bound to the actin. The ComingOffRate influences how long it takes an end of the protein to detach itself from the site where it is bound, where the related ATP has become ADP and results in the end of the protein aiming to bind to another site. If this value is too low, it will result in the ends of the protein being bound at the sites for an inordinate amount of time. Obviously, this is not ideal for the simulation as we aim to decrease the time it takes to travel 30 units as much as possible. However, by increasing this parameter by too much, we run the risk of the protein unbinding at both ends, which can lead to a failure for the protein remain attached at a site while performing the walk. The python code has a threshold where it will automatically give an output when the protein has strayed too far from the actin filament, but the aim is for the protein to never be fully detached from the actin filament – with it being attached to at least one binding site all the time. We made the ComingOnRate 100 times smaller than the original and made the ComingOffRate 6 times larger than the original during our best time trial.

- The VActinDepth parameter refers to the strength of the binding between the protein end and the actin. It affects the depth of this well, and thus influences the amount of energy (potential) for the protein end to be displaced. We increased this parameter by 20 ($30 \rightarrow 50$) so that the binding could be sturdy enough to withstand the other changes to parameters that would cause the protein to fly off completely from the actin filament.

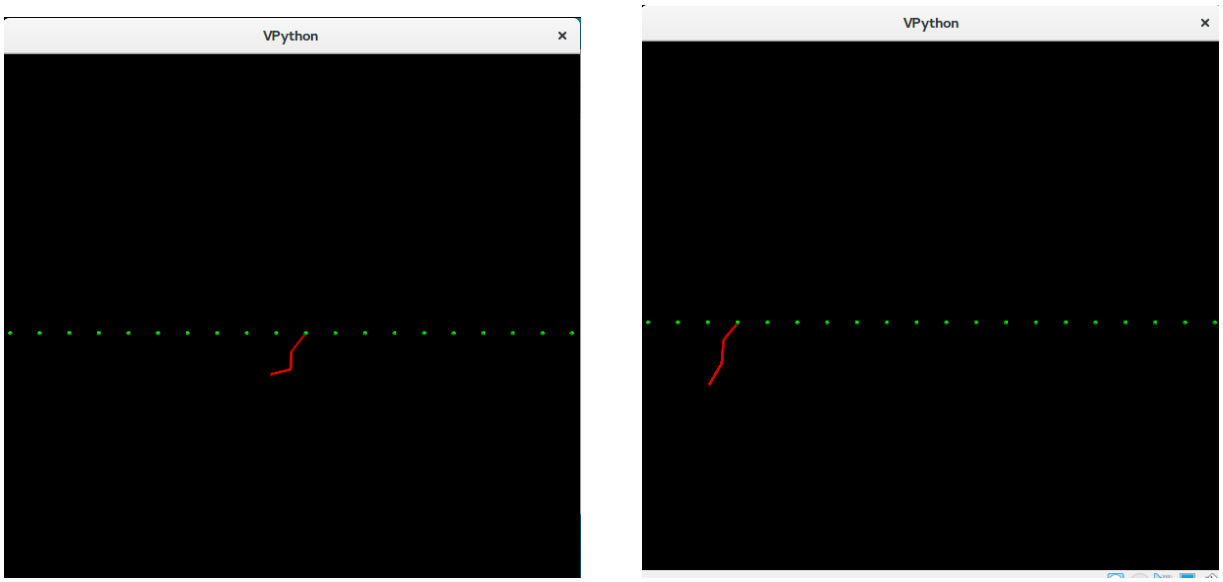
- The Binding Angle parameter is simply the angle at which the protein targets the binding site in order to attached itself. We decreased this angle from $\frac{\pi}{4}$ to $\frac{\pi}{12}$ in order to decrease the angle at which the protein searched around in the medium before settling onto another binding site.

- StepLength has to do with the length of a link and the ChainLength is the sum of the number of links plus one. This allows the addition or subtraction of the number of monomers. We increased the StepLength from 3 to 4 and decreased the ChainLength from 4 to 3. Though the overall length of the protein did not change, what this did was allow there to be just one elbow or joint in the protein, which then allowed it to advance by skipping one binding site during each march ahead. This was a huge time-saver, as I saw in other trials the protein would spend a lot of time just to advance to just the

next binding site and only sometimes skip one. Changing the parameters to these values guaranteed the skip.

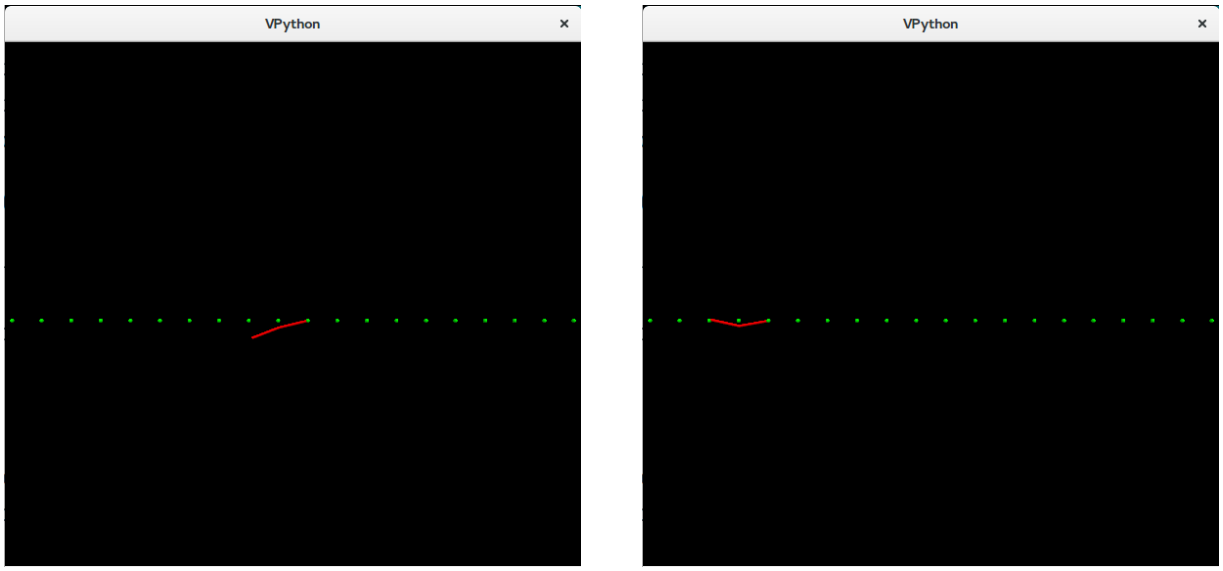
- Lastly, the SpringConstant and the SpringConstantArray dictate the stiffness of the chain, where each monomers stiffness can be adjusted using a function that assigns the spring constant value to each of the monomers. In order to get a chain that is stiffer, the value must be made more negative because the stiffness is regulated by adding more repulsive interactions between the monomers separated by two links. If the stiffness is made too large, then the protein will be two straight and it will be unable to bind at two sites on the actin filament. If the stiffness is too small, on the other hand, the protein will remain in the medium for longer due to having more freedom for backwards movement. We decreased the original SpringConstant value from -0.5 to -0.45, but there was no notable difference when changing this parameter in this range as it still gave us our best time.

By changing the values of a few parameters, we were able to decrease the time from 41.295 to 3.735! Surely there are other methods for decreasing the time that we did not explore, such as going another route and increasing the monomer length or amount, as well as binding angle, and perhaps decreasing the potential actin depth. However, we think we did a respectable improvement on the protein walk along the actin filament.



(a) (b)

Figure 1: Simulation output when run with the default parameter values. The final time is 41.295. (a) An intermediate picture taken as the protein is walking along the actin filament. (b) The final capture of the protein as it reaches the 30 unit "finish line."



(a) (b)

Figure 2: Simulation output when optimized using our changed parameter values. The final time is 3.735. (a) An intermediate picture taken as the protein is walking along the actin filament. You can clearly see the disparity in the chain composition between the original and our altered one, as this one is much stiffer. (b) The final capture of the protein as it reaches the 30 unit "finish line."

Related Biology Questions

If we wanted to construct a motor protein that rarely fell off, with that meaning that it would take a lot of steps binding its ends on the actin filament before accidentally detaching, the speed would be greatly decreased according to the model presented in the code my.py. This is because in order to prevent the protein from completely detaching from the actin binding site, the most important factors were the rates from which the protein comes both on and off the filament. When the rate that the protein would come off was too high, it was common to see the protein flying off into the medium and far away from the filament, which is obviously the opposite of what is desired in this scenario. When lowering this rate though, the trade-off was that the protein would take significantly longer to undergo the walk of 30 units. Increasing the potential energy needed in order to break from the binding site would also ensure that that once the ends of the motor protein would bind to the actin filament, it would not separate so easily and increase the chances of unbinding at both ends. However, doing so also increases the time spent and thus decreases the speed. Considering the design of a motor protein that has a large stall force, with the stall force being a measurement of the strength of the motor protein and the force required to stop it's forward motion if applied in the opposite direction, it would also decrease the speed. If the protein is designed to have a high stall force, it means that the unbinding rate will lower, and thus the protein will be unable to progress as quickly across the binding sites.

Myosin-V was the first myosin discovered as processive, taking multiple steps on its track without dissociating. It starts with both heads in an ADP bound state and the trailing head releases ADP at approximately 12 1/sec which is distinguished as the rate limiting step in the cycle. ATP then binds and dissociates the trailing head which allows the constrained leading head to complete its powerstroke, hence releases stored elastic energy. Next, the detached head hydrolyzes ATP, then re-primes its lever arm to a pre-powerstroke position. It then binds to the next actin binding site as the new leading head, where the phosphate release is fast, and the molecule will be back at the starting position with ADP bound to each head. Several features optimize this processivity: one head must remain bound to the actin filament at all times, meaning that the motor domain has a high duty cycle, and spends most of its time attached to actin in a strong binding state, due to the rate limiting step in the ATPase cycle. Another feature is a high affinity for actin in the

weak-binding states, enhancing the ability of the new leading head to find its next binding site. Also a key feature is that strain-dependent changes in the kinetics of the two heads are happen due to the long lever arms that join the two motor domains, coordinating the two heads in order for their kinetic cycles to be out of phase. Intramolecular strain can accelerate the rate of ADP release from the rear head, inhibit the rate of ADP release from the lead head, or both. Most importantly is to prevent the leading head from dissociating too early. In myosin-V it is apparent, that strain-dependent kinetics are needed for enhancing processive run length (K. Trybus, 2008). Since this motor protein is dependent on a certain rate of movement, especially when transporting RNA, DNA, and crucial information within the cell it would be detrimental if it were slower than the specific rate at which it moves. Unbinding needs to happen quickly and at a lower energetic cost, therefore a large stall force would slow it down or even stop it, and energy required to keep the movement along the actin would be immense and need extra resources. This would not be efficient, and therefore a smaller stall force is optimat for speedy movement and uncoupling.

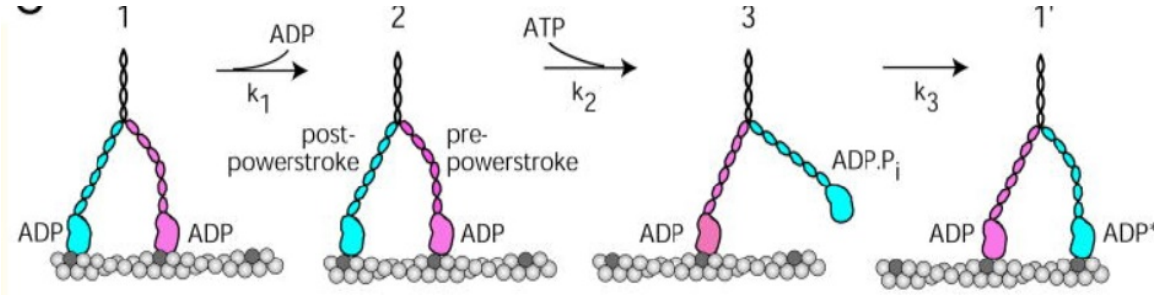


Figure 3: The movement of the arms in Myosin-5 through the ATP-ADP energetic exchange and release from the actin filament (K. Trybus, 208).

2. Cytoplasmic Streaming in *Drosophila* Oocytes

Simulation

During the development of many animals and plants including the widely studied *Drosophila* oocyte (fruit fly egg), there is a phase in which the viscous cytoplasmic fluid and ingredients encapsulated in the egg undergoes fast cytoplasmic streaming. This has shown to be a vital component of proper egg polarization and formation. The fast streaming process is thought to be a result of a force transfer from kinesin motor proteins walking along the array of slender microtubule filaments. This force transfer to the microtubule and the resulting fluid causes the microtubules to bend and form self-correlated arrays, accomplishing widespread motion of viscous cytoplasmic fluid. The theory that this kinesin and microtubule relationship can cause fast streaming has been proposed and experimented on by many groups including Laura Serbus et al. (2005). Through a set of inhibitory and mutant experiments, Serbus et al. were able to determine that the kinesin in necessary to get fast streaming. Joshua Deutsch et al. (2012) continued to study this theory numerically by simulating fast cytoplasmic streaming dynamics through the use of the fundamental physical and hydrodynamic principles involved in this viscous, microscopic system. The main highlights of this model are as follows:

- Kinesin carries cargo that acts as impellers to the surrounding fluid as they are walking along the microtubule from the outside wall (cortex) towards the inner end. The microtubules are in contact and tethered to the cortex (cortex area) at their minus end.
- The impellers are able to affect the velocity of the fluid far from the microtubules because hydrodynamic forces are long range. A relatively small fraction of impellers can cause a large streaming velocity.
- Newtons third law says that for every action (force) there is an equal and opposite reaction (force). This means that the upward motion of the kinesin complex exerts a downward force on the microtubule.

Assuming many microtubules, this can be modeled by saying there is a tangential downward force to the local direction of the microtubule (figure).

- Since the microtubule is tethered at its minus end, this downwards tangential force opposing the direction of walking kinesin proteins will cause a buckling-like instability to form, permitting wavelike motion of the microtubule.

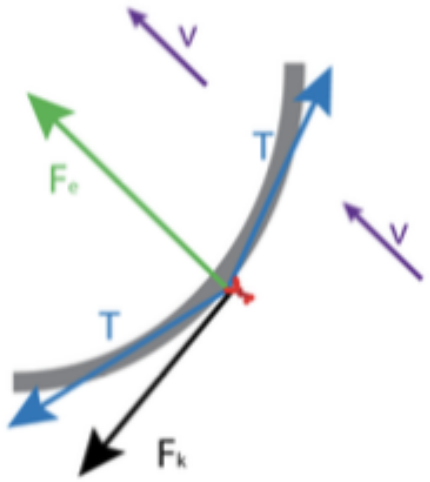


Figure 4: Force diagram for a microtubule and kinesin-1 system. Shown is a section of a microtubule (*gray*) with an attached kinesin-1 motor protein (*red*) and the forces (*arrows*) acting along its length which cause microtubule bending. Kinesin-1 exerts a force tangential to the long axis of the microtubule ($\mathbf{F}_k$). The microtubule has an elastic bending constant that produces a force ($\mathbf{F}_e$) which opposes the microtubule bending. Tension forces (T) act on neighboring components of the microtubule in opposite directions tangent to the long axis at the point where kinesin-1 lies. Force is transferred from the microtubule to the surrounding fluid, which moves at velocity v (Monteith et al., 2016).

The script egg.py simulates the unique instabilities and behavior of microtubules with kinesin along them. Adjusting certain system parameters changes the motion of the microtubules in different ways:

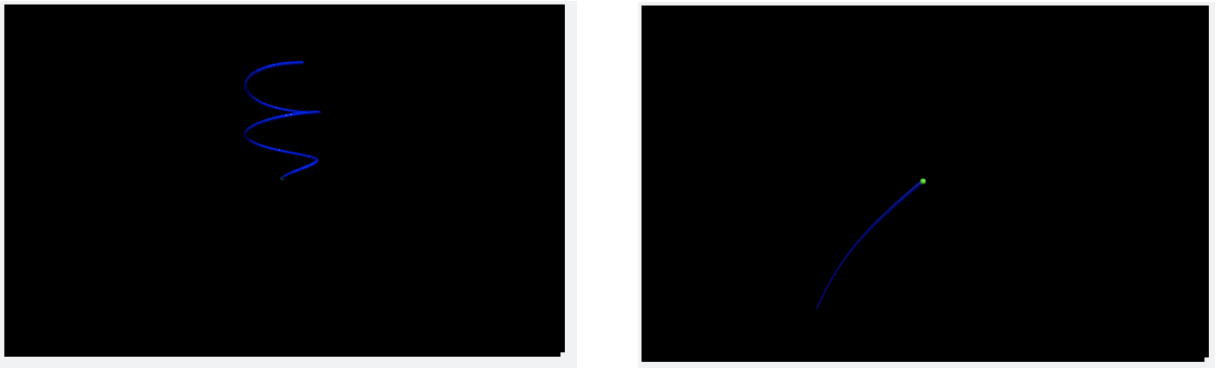
- The number of beads, n, that makeup the microtubules. The dynamics can become more cohesive but the wavelike motion takes longer to develop.

- The kinesin constant, fk, is a measure of how much force the kinesin is applying tangent to the direction of the microtubule. The value in the code is negative which describes the kinesin walking towards the free floating end of the microtubule as in experiments. Therefore a positive value of fk represents a kinesin force acting in the opposite direction. This would make sense if the kinesin were walking in the opposite direction of that which is observed and simulated, i.e. walking towards the tethered base of the microtubule. We can see in figure the difference in the steady state results of the simulation for negative and positive fk. For fk = -1, the microtubule forms a steady state helix wave. This is in agreement with experimental results. The force from the kinesin walking is pointing towards the microtubules tethered base, resulting in a buckling-like motion of the microtubule, which causes it to form a helix pattern in steady state.
  If fk = 1, the force from kinesin is now pointing towards the free floating end of the microtubule. Since this end is not tethered, we would expect the force to simply pull and stretch out the microtubule such that it is a straight line. This is what we see from the simulation results.

Conceptually we can understand these results for both fk pointing up (negative fk) or down (positive fk) the microtubule by comparing this scenario to someone holding a cord at the top end with the bottom end hanging free. Pushing tangentially with a finger towards the tethered end being held results in the cord coiling around itself, forming the helix pattern we see. If we push with a force tangentially down the cord, we see that it just acts to pull the cord out so that it is a straight line.

- The elastic constant of the microtubule, C, can be adjusted. Increasing this stiffness constant does result in the obtainment of fast streaming but substantially delayed its attainment and reduces the associated fluid velocity. This is not ideal in terms of a timely fast streaming process that is effective at moving the fluid far from the microtubule. An intermediate value for StiffnessConst was found to result in the obtainment of correlated microtubule behavior and high fluid velocity.

- The force due to an external velocity field, f_const, can also be adjusted. This quantity is a mean field way of modeling the effects of all the other microtubules with kinesin driven impellers. These produce a streaming velocity field that acts on each other. Turning this off (f_const = 0), is equivalent to simulating a single microtubule with kinesin in isolation (figure).



(a) (b)

Figure 5: Steady state results of cytoplasmic streaming simulation for (a), fk = -1, and (b), fk = 1. For both simulations, the stiffness constant, C = 10, n = 64, and f_const = 0.4. We see that for a tangential force pointing towards the microtubules connected end (a), the resulting pattern is a steady state helix coiling pattern. For a tangential force pointing towards the microtubules free floating end, the resulting dynamics are not as intriguing. This force pulls the microtubule away from its base and so acts to stretch it out in a straight line.

Dimensionally, the elastic constant, C, has dimensions [Force][Length^2], where as the kinesin force constant, fk, has dimensions [Force]/[Length]. By using a combination of C and fk, we can come up with a constant with the dimensions of [Length]:

$$\frac{C}{fk} = \frac{\text{[Force][Length}^2\text{]}}{1}\frac{\text{[Length]}}{\text{[Force]}} = \text{[Length}^3\text{]} = R^3. \rightarrow R \propto (\frac{C}{fk})^{1/3} \tag{1}$$

This tells us that the radius of curvature scales as $C^{1/3}$: $R \propto (\frac{C}{fk})^{1/3}$. Let us set the external force field, f_const = 0 to model the dynamics of a single microtubule with kinesin walkers. We run the code for different values of StiffnessConst (C) and can compare its relationship to the steady state radius of curvature in the middle of the chain (figure).
For C = 10, we find that R = 4.6124.
For C = 20, we find that R = 5.8124.

We can compare these ratios to see how well the scaling factors match:

$$C = 10 :\rightarrow \frac{R}{C^{1/3}} = \frac{4.6124}{10^{1/3}} = 2.1409. \tag{2}$$

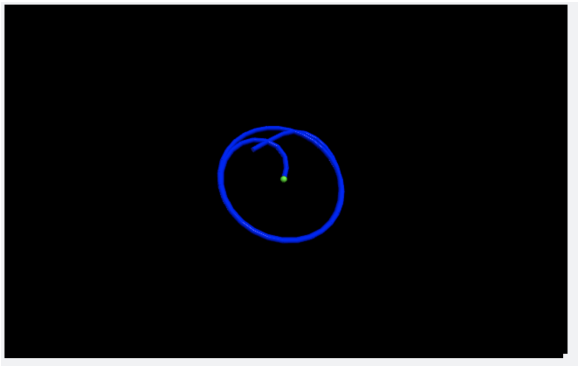


Figure 6: Steady state results of cytoplasmic streaming simulation for f_const = 0.0. This means that we are simulating the dynamics of a single microtubule with the tangential force from kinesin walkers in isolation from any other microtubules. The dynamics of the microtubule now consist of it coiling around itself. We can think of this a a 2d helix, i.e. a helix that is flattened onto a surface. There is no external force from any other microtubles acting on this isolated microtuble so instead of stretching itself out in a 3d helical fashion, it coils over itself in a 2d coiling/helical fashion.

$$C = 20 :\rightarrow \frac{R}{C^{1/3}} = \frac{5.8124}{20^{1/3}} = 2.1413. \tag{3}$$

Taking the ratio of these two values we find:

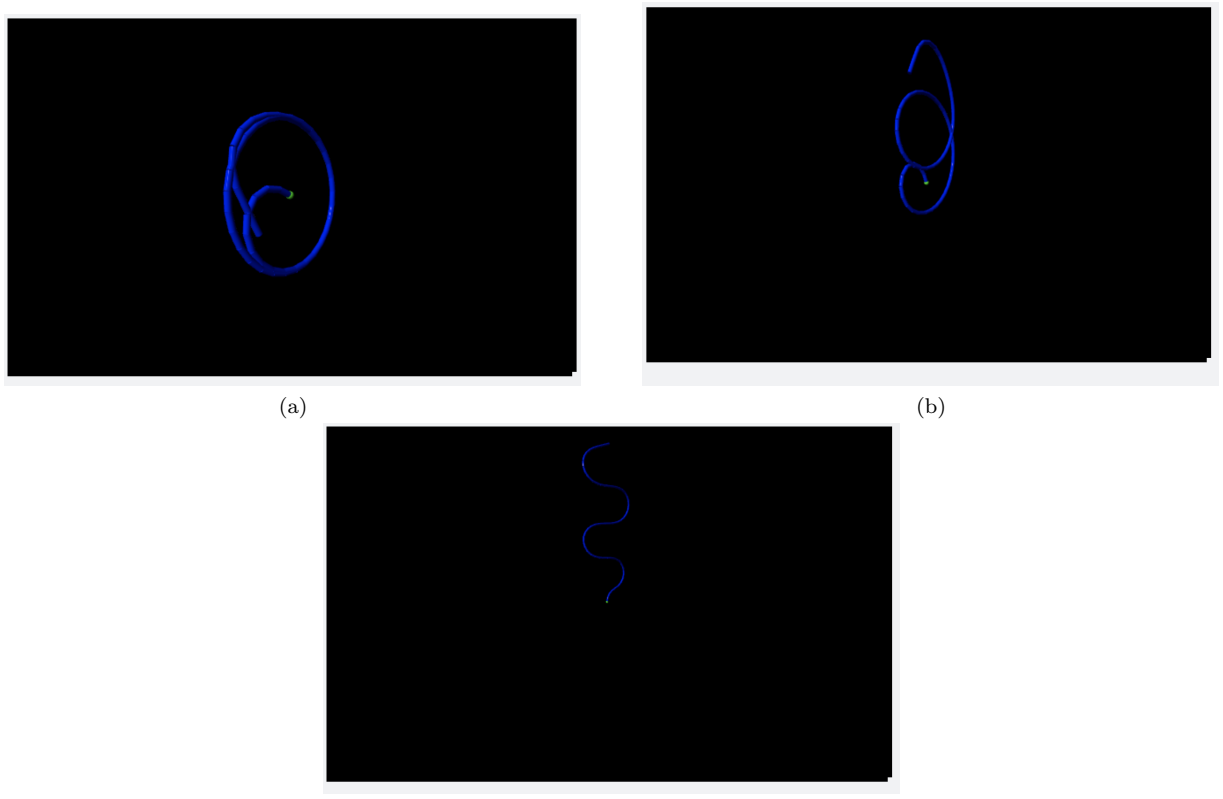
$$\frac{2.1409}{2.1413} = 0.9998 \tag{4}$$

which shows that these two values differ by only 0.019%. This is a small difference and shows these results agree well with the dimensional analysis and scaling. Another way to show this is through the ratio of the two proportionalities:

$$\frac{4.6124}{5.8124} \propto \frac{10^{1/3}}{20^{1/3}} \rightarrow 0.7935 \approx 0.7937. \tag{5}$$

We see that both the left and right hand sides equate nicely up to the 4th decimal place, giving us the same result, which is that these two values differ by only 0.019%.

We can also simulate a cortex wall by adding a wall force and then run the program for f_const = 0,0.3,0.9 (figure). The wall force prevents the microtubule from penetrating a wall located at the plane x = -1 at which the fluid velocity must equal zero. We see in figure the microtubules dynamics for different values of f_const. What we get is essentially the 2d version of the previous microtubules dynamics without a wall, where now the microtubule moves parallel to the wall. When there is no force from the external velocity field (f_const = 0), the microtubules loops back around itself in a circular coiling fashion parallel to the plane wall. If we increase the force from the external velocity field to f_const = 0.3, we see that the steady state travelling wave solutions are still parallel to the wall and are similar to a cycloid shape, that which periodically loops back over itself. If we increase f_const = 0.9, we find that the 2d motion of the microtubule-like filament becomes sinusoidal, similar to fast streaming microtubule motion experimentally observed near the cortex wall of the oocyte. We see that for a larger external velocity field force, f_const, the microtubule motion gets pulled out. For f_const = 0.9, the filament does not loop over itself anymore but rather moves in a sinusoidal like fashion parallel to the plane wall.
Ultimately, because the external velocity field force still runs in the same direction, parallel to this added wall force, the resulting filament dynamics flow parallel to this wall as well. But they are confined to 2d movements because the zero velocity boundary condition at the wall prevents the filament from making three dimensional movements like the helical pattern which the filament undergoes without the wall force.



(a) (b) (c)

Figure 7: Simulated cytoplasmic streaming dynamics after a plane at x = 1 has been added for (a), f_const = 0.0, (b), f_const = 0.3, (c), f_const = 0.9. We see that the microtubule still undergoes similar dynamics as for the simulations without a wall, but now the microtubule is essentially performing these dynamics in a 2d fashion, parallel to the plane (wall).

Cytoplasmic Streaming Biology Questions

One experiment that Serbus et al. did was to inhibit the dynein motor protein before the fast streaming stage (2005). Dynein move along the microtubule but in the opposite direction as kinesin. Therefore it produces a tangential force opposite the direction of kinesin's tangential force. Inhibiting this tangential force which is opposite that of kinesin's means that we would expect this dynein inhibition to speed up the obtainment of fast cytoplasmic streaming. This is exactly what their data and experiments show. Essentially it is easier for walking kinesin to make the microtubules buckle and produce fast streaming if there is less or no opposing tangential force on the microtubule from the dynein motor protein. These experiments suggest that dynein can be or is a mechanism of inhibiting kinesin driven fast streaming and preventing it from happening too early. Premature fast streaming can be lethal to a developing oocyte which is why there are mechanisms such as dynein or the f-actin cytoskeleton which have been found to repress fast streaming during early and mid oogenesis. Experimentally Sebrus et al. have found agreement within this theory given that dynein is found to be suppressed just before nurse cell cytoplasm is dumped into the oocyte in preparation for fast streaming (2005). This allows kinesin to apply a strong enough minus-end-directed tangential force along the microtubule, allowing the slender filaments to form parallel arrays that enhance long range and fast fluid motions.

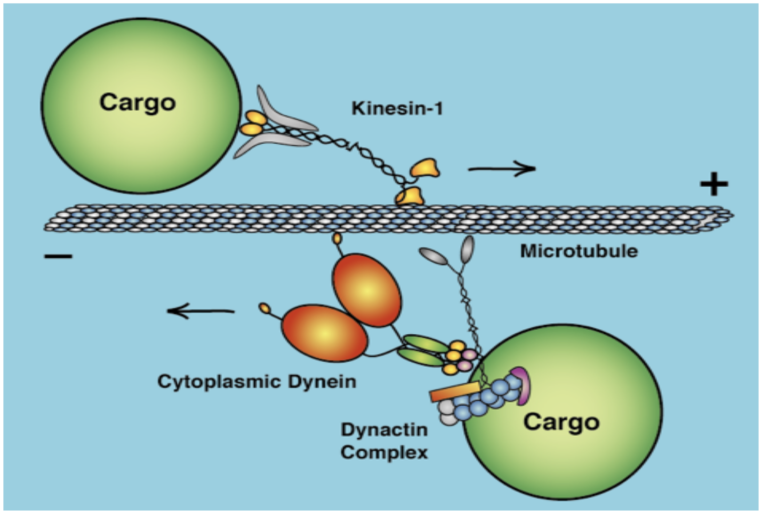


Figure 8: Schematic of a microtubule with cargo-carrying kinesin-1 and dynein motor proteins. Microtubule is made up of tubulin, which is a dimer consisting of alpha-tubulin and beta-tubulin, represented in the figure as blue and white spheres respectively. Kinesin-1 and dynein walk in opposite directions along the microtubule (Monteith, 2016).

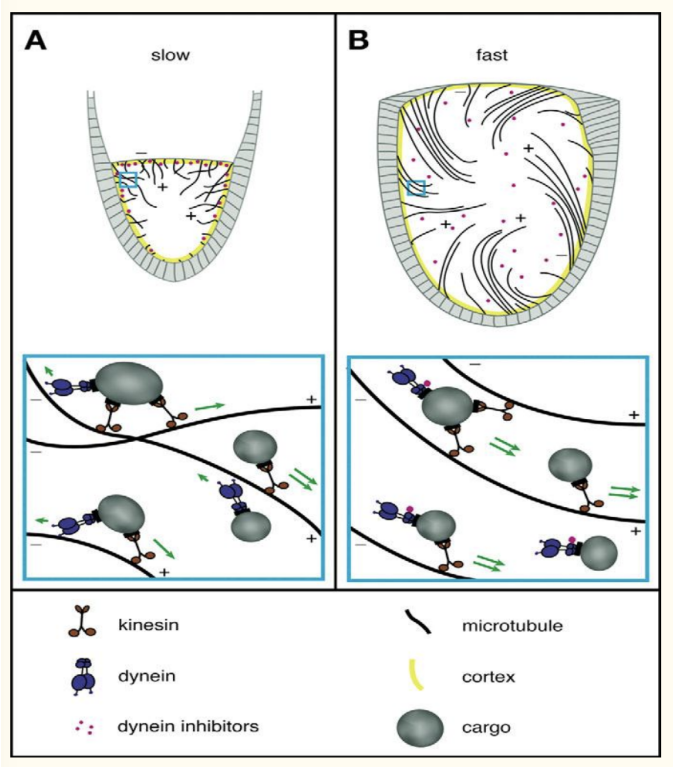


Figure 9: A schematic showing the differences between slow and fast streaming and what can cause this. The underlying idea is that the presence of the kinesin-opposing dynein motor protein can act to suppress fast streaming, allowing only slow, short ranged, and disordered cytoplasmic streaming (A). During the fast streaming stage (B), a dynein inhibitory signal is released, allowing the kinesin-driven cargo transport to sweep the microtubules into parallel, self organized arrays that which can accomplish fast and long ranged cytoplasmic streaming (Serbus et al., 2005).

When Serbus et al. loosened the strength of the actin cytoskeleton before the fast streaming stage, this produced a similar affect as the inhibition of dynein; inducing premature fast streaming (2005). Serbus et al. find that the depolymerization or mutation of certain actin-interacting proteins can induce premature kinesin driven fast streaming (2005). The idea is that the gel-like actin mesh can help assist in preventing premature fast streaming because it is passively increasing the cytoplasm viscosity in addition to possibly generating antagonistic forces against those produced by kinesin. If this actin mesh is depolymerized, this allows the kinesin driven forces to dominate and produce fast streaming.

Serbus et al. determined that kinesin was necessary for fast streaming through multiple kinesin inhibitory and mutant experiments. Their results show that the partial inhibition of kinesin by two hypomorphic Khc alleles either blocked or severely reduced fast streaming (Serbut et al., 2005). The resulting streaming patterns were mainly short, individual displacements, with only one out of the nine test oocytes showing concerted and localized fast streaming currents (2005).

When cytoplasmic streaming is suppressed, the oocyte's development is severely hindered and oocyte fatality is possible. The oocyte is not able to correctly localize and polarize its DNA, or mix its components thoroughly without fast streaming. If cytoplasmic streaming is suppressed, as found in Khc-null oocytes with no streaming, yolk stratification is observed as evidence of mixing failure (2005). These oocytes then led embryos which ceased to continue developing in early stages. Ultimately, this suggests that the cytoplasmic streaming mixing mechanism is important for proper subsequent oocyte and embryotic development.

3. Image Processing

Image processing is often used on raw data from many kinds of microscopy. Fourier transforms are of great use in image processing. The transform is used to decompose an image into its sine and cosine components and so the output of a Fourier transform represents the raw data or image in the frequency domain. Each point represents a particular frequency contained in the spatial domain image.
The image, sampling_image.png is a bunch of wavy tubes that have been blurred and has noise added. For example, this might be an image of DNA chains or a substrate.

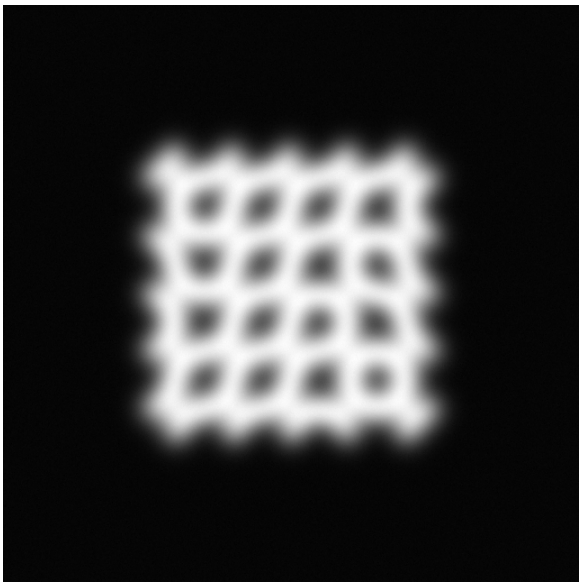


Figure 10: Sample image of a bunch of wavy tubes which have been blurred and has noise added.

The goal here is to process or filter the image (in k, or frequency, space) well enough so that we can distinguish individual tubes from each other.
It is worth mentioning the idea of a deconvolution which can be used to deblur the image. In deconvolution, two functions are divided in the Fourier (frequency) domain to recover the original image with greater clarity; a popular technique in spectroscopy from an empirical standpoint. Because the image we are dealing with is made from discrete data points, a Discrete Fourier Transform can be used to perform certain types of image deconvolution.
However in the case of this problem it is not completely necessary to perform a deconvolution. We can still filter the image in k space such that we can distinguish the individual tubes in the image even if the image is not crystal clear. A high pass filter will suffice in clearing up the image. This type of filter will eliminate some of the low frequency noise and blurryness present in the image and will return a much clearer image, although solely a high pass filter will not be able to make the image crystal clear. In brief, we take the fourier transform of our image to bring it to the frequency domain. We then apply a high pass filter to the image. Lastly we take the real part of the inverse fourier transform of this new image (and make sure the image is normalized as well) so that we can see the resulting, deblurred image.
We can think of a high pass filter equation like we would in circuits and how we can make a high pass filter with a capacitor and resistor. Let us then relate this high pass filter to the voltage going out divided by the voltage coming into a circuit, $\text{filter}_{\text{hp}} = |V_{\text{out}}/V_{\text{in}}|$.
This voltage ratio is the transfer function that which looks like a high pass filter. In terms of the program we are running in reconstruct_image.py, (ksq) represents the image location points in frequency space that is our output, and (ksq + cutoffsq) represents our input in frequency space, where cutoffsq is the cuttoff

frequency we choose squared. Therefore we chose to model this high pass filter with the equation:

$$\text{Filter}_{\text{hp}} = (\frac{\text{ksq}}{\text{ksq} + \text{cutoffsq}})^n \tag{6}$$

where n is a constant that determines how steep i.e. sharp this filter is. After testing multiple values it seemed that the best value for n was $n = 2$. This made for the clearest resulting image after this high pass filter at each point in frequency space was multiplied by the fourier transform of the original image at that same location. Additionally the best value for cutoff seemed to be between the range of 15 and 20. We chose to render the resulting image using cutoff = 18. If the value of cutoff was too large, the image was too hard to see as we are not letting in enough of the low frequency range. If the value of cutoff is too low, there is too much low frequency spectra in the image and the resulting image is still too blurry. A nice middle ground high pass filter cutoff value was 18. This cut out enough (and not too much) of the low frequency spectra which allows us to see a resulting image that is less blurry, as seen in figure . Utilizing such a high pass frequency filter allows us to distinguish the individual tubes in the image! Although we see the image is still slightly blurry, it is much more clear than the original image.

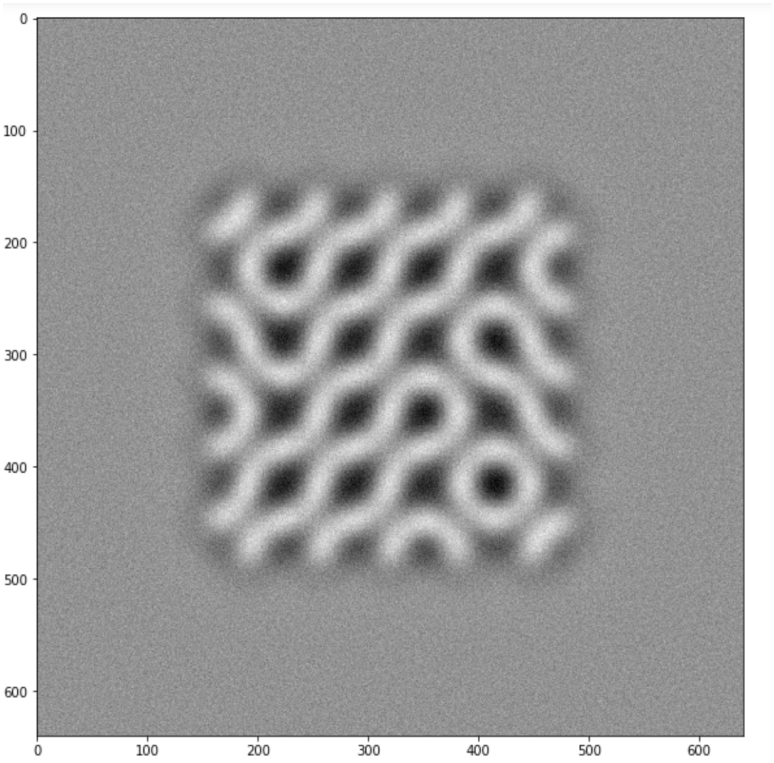


Figure 11: The resulting deblurred image after a high pass filter was applied in frequency space. A frequency cutoff value of 18 was used.

Image Processing: Practical Problems

Some practical problems that can or most likely will happen with image analysis are such as noise, due to light intensity or lack thereof in microscopes, it has been found that florescence images can be contaminated by the autofluorescence. Then there is the issue with either low resolution and or low frame rate, as well as blurring. Characteristics of specific target objects can prove to be difficult, for example having less appearance information, mainly showing a bright spot or very small object. Photobleaching of dying cells can bring about unstable brightness, and very important are the presence of multiple targets, overlapping targets, transparent objects and unclear boundary or texture (S. Uchida, 2013).

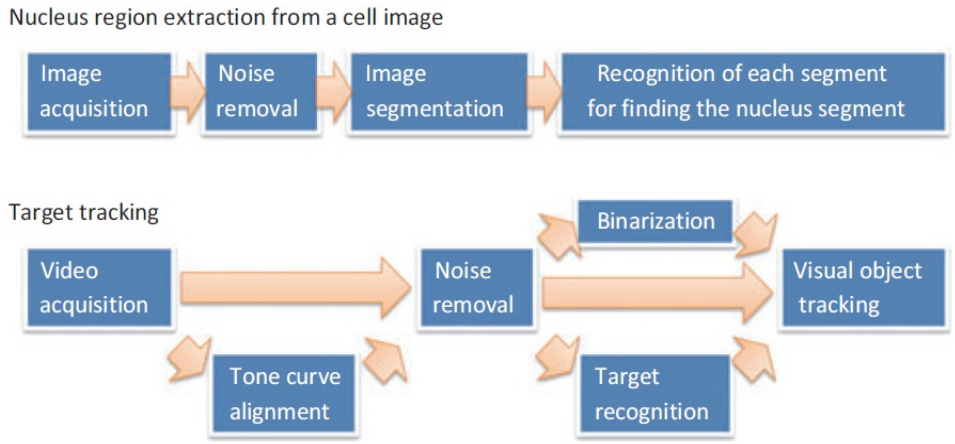
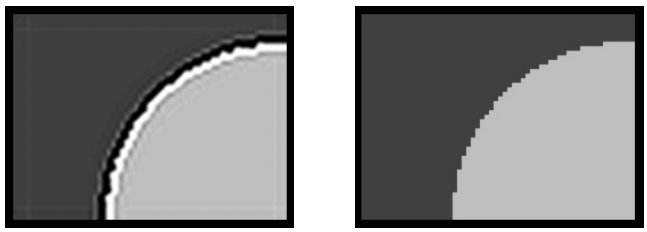


Figure 12: Multiple image processing and recognition techniques are necessary in implementing a well-rounded system for specific imaging tasks (S. Uchida, 2013).

Ringing artifacts can be described as when the output takes on values higher than the maximum or lower than the minimum input value. These can occur one without the other, yet with a low-pass filter, we first have overshoot and the response bounces back below the steady-state level. This will cause the first ring, then oscillates above and below the steady-state level. Next, ringing will be the second and subsequent steps. A linear time invariant filter can be used to understand the filter and ringing in terms of the impulse response, known as the time domain view. There is also the frequency response or frequency domain view, in terms of its Fourier transform (Wikipedia.org, 2021). When looking at biological images, ringing can be a real issue and fake results will highly impact the understanding of any boundaries or defined contours, crucial to recognition and understanding of what is viewed and how it functions. When working with subject matter that is not visible with the naked eye, this is a big problem, therefore filters and ringing artifacts have to be corrected as efficiently as possible. The veracity of the image is highly impacted by this imaging problem, and this impedes proper identification, further research, correlation to biological systems, and cataloging of biological data.



(a) (b)

Figure 13: (a) Showing three levels: overshoot first ring and second ring of ringing artifacts around the boundary of this image. (b) The boundary without ringing artifacts (Wikipedia.org, 2021).

Undersampling, known as aliasing artifact, refers to an error in the accuracy of analog to digital conversion during image digitization, which has three distinct steps: scanning, sampling, and quantization. During sampling, each pixel brightness is measured, creating an output analog signal due to undergo quantization. The more samples are taken the higher the accuracy of the representation of the signal. Therefore, if a lack of sampling has occurred, it will be processed as an inaccurate image resulting in an aliasing artifact, which has the appearance of moire patterns (radiopedia.org, 2021). The moire effect occurs when liens or dots are viewed that are superimposed on other lines or dots, and these sets differ in size, angle and or spacing. It can be generated deliberately or accidentally, and can produce geometric patterns, but it will degrade the resolution of images. Sometimes, the matrix of dots in the original image almost invariably conflict with the matrix of dots in the reproduction. This generates a criss-cross pattern on the reproduced image (whatis.techtarget.com, 2021). There are four methods that help remove aliasing in standard images or screens: high-resolution display, applying post filtering or supersampling, using pre-filtering or area Sampling, and pixel phasing. For biological imaging, we use Fourier transforms, convolution theory, and sampling theory to deduce the real from the aliasing and moire patterns and interference.

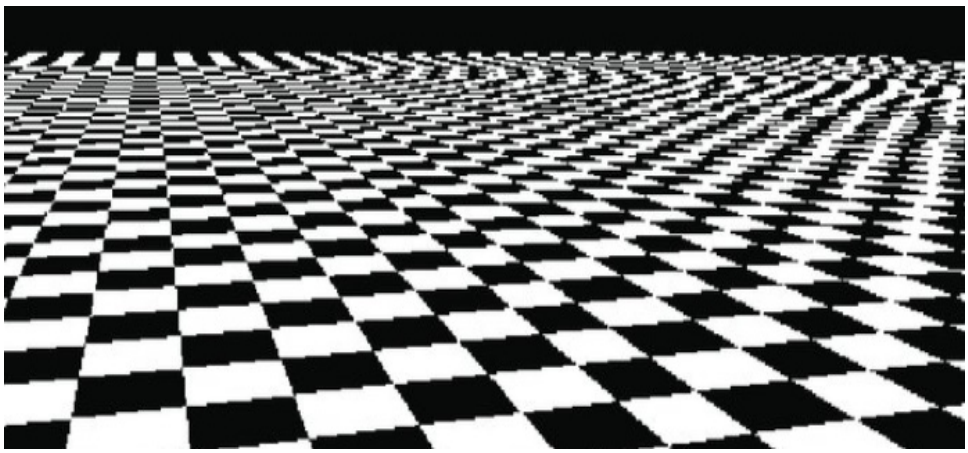


Figure 14: Typical aliasing artifact similar to the moire effect. (Researchgate.com, 2021).

Scanning tunneling microscopy (STM) has to discern not only contours but also depth in the images. This can be very deceptive when encountering artifacts in the substrates and the recognition or separation between the substrate and the object of interest is hard to distinguish. Removal of artifacts will help in the recognition of these dimensions and separate substrate from the desired imagery. It is crucial in research to have a clear image without interference, therefore these types of artifacts can be very misleading when not taken into account.

Sources and References

Monteith, C. E., Brunner, M. E., Djagaeva, I., Bielecki, A. M., Deutsch, J. M., Saxton, W. M. (2016). A Mechanism for Cytoplasmic Streaming: Kinesin-Driven Alignment of Microtubules and Fast Fluid Flows. Biophysical Journal, 110(9), 2053–2065. https://doi.org/10.1016/j.bpj.2016.03.036

https://radiopaedia.org/articles/aliasing-artifact-ct?lang=us

https://www.researchgate.net/publication/3410959 EWA splatting/figures?lo=1

Serbus, L. R., Cha, B.-J., Theurkauf, W. E., Saxton, W. M. (2005). Dynein and the actin cytoskeleton control kinesin-driven cytoplasmic streaming in Drosophila oocytes. Development, 132(16), 3743–3752. https://doi.org/10.1242/dev.01956

Trybus, K M. "Myosin V from head to tail." Cellular and molecular life sciences : CMLS vol. 65,9 (2008): 1378-89. doi:10.1007/s00018-008-7507-6

Uchida, Seiichi. "Image processing and recognition for biological images." Development, growth differentiation vol. 55,4 (2013): 523-49. doi:10.1111/dgd.12054

https://whatis.techtarget.com/definition/moire-effect

https://en.wikipedia.org/wiki/Ringing artifacts