Supporting Text

Systematic Error in Velocity. Single-particle tracking measurements have a systematic error, which causes the average apparent distance that a particle has moved to be larger than the actual distance.

This effect becomes more pronounced because the displacement decreases between successive positions. Here, we will derive the expression for the systematic error in the measurement of displacement in single-particle tracking. As far as we know this phenomena is not discussed, but it may have some consequences for high-accuracy tracking experiments. Referring to Fig. 8 and choosing a coordinate system centered at the left circle, all points contained by the two circles are given by

$$\vec{p}_1 = (r_1 \sin\theta_1, r_1 \cos\theta_1) \qquad \textbf{[1.1]}$$

$$\vec{p}_2 = (R + r_2 \sin\theta_2, r_2 \cos\theta_2) \qquad \textbf{[1.2]}$$

where 1 refers to the left circle and 2 the right, $r_{1,2} \in [0, \delta R]$ and $\theta_{1,2} \in [0, 2\pi]$. Then, the average distance between two randomly selected points is given by integrals over all possible left and right ends. Defining $\bar{D}$ to be the average distance, the integral is

$$\bar{D} = \frac{1}{\left(2\pi\delta R^2\right)} \int_0^{2\pi} d\theta_1 \int_0^{2\pi} d\theta_2 \int_0^{\delta R} dr_1 \int_0^{\delta R} dr_2 r_1 r_2 \left|\vec{p}_1 - \vec{p}_2\right| . \qquad \textbf{[1.3]}$$

This integral is not solvable in closed form; however, we can pull the *R* out of the integrand and Taylor expand in terms of $\frac{\delta R}{R}$, which is hopefully not too large or else other problems will be dominant. When expanding, all terms that contain odd powers of sine or cosine can be ignored because the integral over a full period will be 0. To second order, the contributing terms are

$$\bar{D} = \frac{R}{\left(2\pi\delta R^2\right)} \int_0^{2\pi} d\theta_1 \int_0^{2\pi} d\theta_2 \int_0^{\delta R} dr_1 \int_0^{\delta R} dr_2 r_1 r_2 \left[1 + \frac{1}{2}\left(\frac{r_1}{R}\right)^2 \cos^2\theta_1 + \frac{1}{2}\left(\frac{r_2}{R}\right)^2 \cos^2\theta_2\right] . \qquad \textbf{[1.4]}$$

The remaining integrals are trivial and the result is

$$\bar{D} = R\left[1 + \left(\frac{\delta R}{2R}\right)^2\right] . \qquad \textbf{[1.5]}$$

To check that this result is not just a problem with the expansion, we have also simulated the situation. These straightforward simulations entail choosing two circles of fixed radii separated by a varying distance *R* to represent the data points and their error. In all cases, we choose the radius $\delta R = 1$ and vary R such that $\delta R / R$ goes from 0 to 1, representative of the relative range. Note that a $\delta R / R$ = 1/2 corresponds to the two circles just touching. Then, the distance between a million randomly selected points in each circle is calculated and averaged. The result shown in Fig. 8 shows excellent agreement between the calculation and the simulation, although the calculation produces a noticeable but slight underestimate as $\delta R / R$ approaches 1.

With this systematic error in mind, we calculate the speed of the cargos by plotting the apparent speeds calculated after time spans of 1, 2, ... , 10 frames, where one frame is 7.5s here. Then, we correct the speed by using Eq. **1.5** and vary δR until the corrected speeds do not depend on the time span of the data. With this calculation, the values for the speeds are 70 ± 20 nm/s with $\delta R = 24 \pm 4$ nm for aggregation and 80 ± 30 nm/s with $\delta R = 20 \pm 4$ nm for dispersion. The values of δR are reasonable for these in vivo systems.

Details of the Simulations. There are two types of intersections: “cross intersections,” where two filaments cross, and “end intersections,” where a new filament is reachable from the end of the old filament (Fig. 10). By using geometry, the computer identifies these intersections as filaments are added during the process of setting up the AF network. Call the newly added filament the “current” filament and all others “target” filaments. After each filament addition, the computer examines all target filaments for both cross and end intersections with the current filament, and vice versa, and, if necessary, it finds the point on the target filament to which a cargo switching from the current filament will go. Each filament is described by an equation for a line segment. The fact that a filament is a line segment is handled by keeping track of the end points. Determining the existence of a cross intersection is done by solving the equations of the two filaments for their common point and then checking whether that common point is on one of the filaments. Note that this also results in the location of the intersection, which is recorded at the same time and used as the transfer point. For end intersections, we consider a circle centered at the filament end point with a radius set to 250 nm to coincide with a cargo. The existence of an intersection is determined by first checking whether the endpoints of the target filament are inside or outside of the current filament’s end circle. If exactly one or two endpoints are interior to the circle, then intersection occurs. If both are exterior, then the perpendicular distance to the target filament is found, and whether it is less than or equal to the end radius, then an end intersection occurs. If an end intersection is found to occur, then the transfer point is defined as the point on the target filament with the minimum distance to the end of the current filament. Note that it is certainly possible, and fairly common, for a target filament to have both an end and cross intersection. The main simplification made here is that the filament network is fixed. To be applicable to real systems where AFs do move, this fixed approximation means that the time scale on which AFs change is long compared to that on which the cargos move. This is a reasonable approximation because the time scale for to AFs change is on the order of several minutes.

Now, we describe the motion of the cargos on the filaments in the simulation. To start, a filament near the center of the simulation cell is chosen at random and a cargo is placed randomly on it. Then, the cargo takes a step of size 37 nm, corresponding to the size of an

MV step (1). First, the computer checks whether the cargo has attempted to walk off the end of the filament, and if so, it randomly selects an end intersecting filament and switches the cargo onto it. Second, whether the cargo remains on the filament, then the code looks for cross intersections on this step. If there are any cross intersections, then the cargo attempts to switch onto each one with success probability $p_{s,}$ set to 50% for aggregation or 0% for dispersion. Any successful switches immediately move the cargo onto the crossing filament. If no intersection is found or accepted, the original step is accepted, and the process is repeated. Also, while this process is going on, the distance between turns is tracked for use in calculating the MFP, and the position of the cargo is tracked for calculating $<r^2(t)>$. Trials are ended after up to 10,000 steps or if the cargo attempts to leave the simulation cell. It is possible, but relatively rare, for a cargo to fall off and be unable to find a new filament on which to travel, and in this case the trial is prematurely ended. Presumably, in the real system, lateral diffusion would eventually allow the cargo to reach a new filament.

MFP Calculation. We give the details of our calculation of the MFP given the distance d_I between filament crossings. Let p_s be the probability to switch filaments at an intersection. Then $q = (1- p_s)$ is the probability to skip switching at an intersection. The probability $P(n)$ to skip $(n - 1)$ intersections and then switch at the nth intersection is given by $P(n) = p_s q^{n-1}$. Assuming that the cargo attaches to a filament at an intersection, the distance that it goes along a filament before switching to a new filament at the *n*th intersection (not counting the intersection where it first got on) is given by nd_I. The average distance traveled along a filament, i.e., the MFP, is given by

$$\begin{aligned}
\text{MFP} &= \sum_{n=1}^{\infty} nd_I P(n) \\
&= \sum_{n=0}^{\infty} nd_I pq^{n-1} \\
&= pd_I \frac{d}{dq} \sum_{n=0}^{\infty} q^n \\
&= pd_I \frac{d}{dq}\left(\frac{1}{1-q}\right) \\
&= \frac{pd_I}{(1-q)^2} \\
&= \frac{d_I}{p}
\end{aligned}$$

This finding agrees with the expression given in the main text. Here, we are assuming that the filament is infinitely long.

Possible Optimization Between Aggregation and Dispersion: MT Searching. For aggregation, the goal is to hop onto a microtubule for subsequent efficient transport to the cell center, so local searching is optimized. Whether this is the case is not directly evident from the current data. To investigate aggregation further, simplified MTs are added to the simulations as parallel lines spaced by 800 nm. A range of distances comparable with the radius (±125 nm) of a pigment granule is defined around each MT within which a cargo is considered to contact the MT. If only a single cargo-MT contact is required, dispersing-type motion is as good or better than aggregation-type motion in finding an MT.

However, if multiple cargo-MT contacts are required for transfer from the AF to the MT, aggregation-type motion is better. This result can be seen by using an arbitrary requirement of 15 contacts. The time required for a cargo to make 15 contacts, or about two passes perpendicularly across an MT, is tracked for 100 realizations with 10,000 cargos each (Fig. 11). In the first 20 s, an aggregating cargo is about 20% more likely than a dispersing cargo to contact any MT 15 times. A dispersing cargo can possibly reach a second set of MTs after 13s, so its rate of contact accelerates, catching up to aggregation after about 20s. Thus, an aggregating cargo contacts the very first MT it reaches 15 times, but a dispersing cargo tends to zip past the first MT and not get 15 contacts until it finds a second MT. Importantly, after a total of approximately 500s, 5.5% of dispersion cases never contact an MT 15 times but only 0.5% of aggregation cases never do. Thus, fewer aggregating cargos are "stranded" on the actin, improving overall transport toward the nucleus. Therefore, aggregation is better than dispersion for allowing multiple contacts between the cargo and the MT and quickly searching locally.

1. Yildiz, A., Forkey, J. N., McKinney, S. A., Ha, T., Goldman, Y. E. & Selvin, P. R. (2003) *Science* **300,** 2061-2065.