Supporting Information

Mortimer et al. 10.1073/pnas.0909254107

Detailed Computational Methods

Neurite Pixel Identification and Orientation Estimation via Steerable Filters. To estimate the degree of turning present in an explant image, it first was necessary to isolate which pixels in a given image corresponded to neurites (hereafter referred to as "neurite pixels") as well as the local orientation of the neurite at the locations specified by these pixels. Both these pieces of information were obtained using the ridge-filtering approach developed by Meijering et al. (1–4). This approach involved assembling, for every pixel location $\boldsymbol{r} = (x,y)$ in the image, the matrix

$$H(\boldsymbol{r}) = \begin{bmatrix} f_{xx}(\boldsymbol{r}) - \frac{1}{3} f_{yy}(\boldsymbol{r}) & \frac{2}{3} f_{xy}(\boldsymbol{r}) \\ \frac{2}{3} f_{xy}(\boldsymbol{r}) & f_{yy}(\boldsymbol{r}) - \frac{1}{3} f_{xx}(\boldsymbol{r}) \end{bmatrix} \quad [1]$$

with $f_{ij}(\boldsymbol{r}) \equiv (f^{*}\frac{\partial^2 G}{\partial i \partial j})(\boldsymbol{r})$, $G(\boldsymbol{r}) \equiv \exp[-|\boldsymbol{r}|^2/(2\sigma^2)]/(2\pi\sigma^2)$ and $f(\boldsymbol{r})$ being the brightness of the pixel at position $\boldsymbol{r}$. The standard deviation σ of the Gaussian was set to three pixels, a value which optimized the response to ridges with a width corresponding to the average width of neurites in the image. For each pixel, the eigenvectors and corresponding eigenvalues of $H(\boldsymbol{r})$ were identified and used to calculate the "neuriteness"—a positive real number between 0 and 1 defined as

$$\rho\left(\boldsymbol{r}\right) = \begin{cases} \lambda(\boldsymbol{r})/\lambda_{\min} & \text{if } \lambda(\boldsymbol{r}) < 0 \\ 0 & \text{if } \lambda(\boldsymbol{r}) \geq 0 \end{cases} \quad [2]$$

with $\lambda(\boldsymbol{r})$ being the largest magnitude eigenvalue of $H(\boldsymbol{r})$ and λ min being the smallest magnitude eigenvalue identified over the entire image. Neurite pixels then were identified as those pixels for which the neuriteness exceeded some threshold value ρ_{thresh} (we found that $\rho_{\text{thresh}} = 0.03$ achieved good segmentation). Neurite orientation angles were identified as the orientation of the eigenvectors corresponding to the smallest (in magnitude) eigenvalues of $H(\boldsymbol{r})$ at these locations.

Explant Simulations. As noted in *Materials and Methods* in the main text, simulating a single explant involved three steps: (*i*) generating a mock explant body and seeding it with neurite outgrowth sites, (*ii*) generating neurite trajectories according to the growth models of interest, and (*iii*) compositing the explant body with the neurite trajectories (with a degree of imaging noise) to generate a final simulated explant image. The methods used to generate neurite trajectories are described in the main text. We now describe the first and third steps.

Generation of the explant body and selection of neurite outgrowth sites. Each explant first was assigned an average radius *r*, selected from a normal distribution with a mean of 120 pixels and SD of 20 pixels. Noncircular explant body shapes were generated by randomly assigning values to the Fourier coefficients of a polar function up to order 5, with constant offset *r*. The amplitude of the *n*th order Fourier component was selected from a normal distribution with mean zero, and an SD $r^{*}a_0/n$. We varied a_0 to achieve visually realistic explant bodies, ultimately selecting $a_0 = 0.13$.

Each explant was assigned an average neurite density ρ per pixel, selected from a normal distribution with a mean of 0.033 and SD of 0.02. Thus, the number of neurites simulated for an explant body of area *M* pixels was given by $n = M\rho$. To allow for hotspots of outgrowth, we first generated a random surface (again by creating a random superposition of Fourier components, this time in two dimensions) and then placed each of the *N* neurite initiation sites on the explant body, with probability of site selection modulated by the random Fourier surface. To account for the observation that only a subset of dorsal root ganglion (DRG) cells expresses the nerve growth factor (NGF) receptor TrkA, each neurite had a 50% probability of being responsive to the gradient.

Compositing explant body and neurite trajectories to form final image. Neurites first were converted from *x*, *y*, *z* coordinates to image coordinates with intensity dependent on the *z* coordinate. The resulting image was blurred with a Gaussian filter. The neurite and explant body images were combined, and the resulting image was blurred further to remove sharp edges from the explant body. Speckle noise then was added to the resulting image.

Clarifying the Meaning of Turning

In the main text, we distinguish between the turning and growth-rate modulation models of axon guidance by asserting that the former model exhibits "immediate and biased turning" in the direction of the gradient. By "immediate and biased turning," we mean that at each timestep, the growth cone assesses the gradient direction and then takes its next step in the direction it believes is closer than its current heading to the true direction of the gradient.

In the growth-rate modulation model, the growth cone is equally likely to turn toward or away from the gradient direction on each timestep. Guidance occurs because each step tends to be longer when the growth cone happens to be growing up the gradient than when it happens to be growing down the gradient.

Does this difference in mechanism lead to quantifiable differences in neurite trajectories produced by these models? Although neurite turning is effectively "built in" to the turning model, it is not immediately obvious whether one will observe turning in the growth-rate modulation model in the long term, when the effects of multiple steps are combined. Indeed, in one sense, both models display what might be called "turning": Fig. S6 *A* and *B* illustrates that in both models, if axons initially project along the positive *x* axis, the average trajectory taken by the neurites turns in the direction of the gradient. It is obvious that this must be the case: Both models cause net progress up the gradient, and because, in this situation, all axons begin by growing horizontally, there must be a turn in the average trajectory. In each simulation, 500 neurites were simulated over 100 timesteps. The gradient strengths were $\delta_{GRM} = 0.5$ for the growth-rate modulation model (Fig. S6*A*) and $\delta_T = 0.05$ for the turning model (Fig. S6*B*).

However, although neurites in the growth-rate modulation model show progress up the gradient, in another sense they do not display biased turning toward it, in contrast with neurite trajectories in the explicit turning model. In Fig. S6 *C* and *F* we have plotted histograms of neurite curvature (transformed on a log-scale*) conditional on the current heading direction of the neurite. Here, as in the calculation of the curvature ratio in Fig. S2, curvature was calculated by smoothing the neurite trajectories using a smoothing spline (Matlab spaps function) and then applying the standard definition of 2D (scalar) curvature:

$$\kappa = \frac{v_x a_y - v_y a_x}{(v_x^2 + v_y^2)^{3/2}}, \quad [3]$$

where v_x and v_y are the *x* and *y* components of the velocity $\partial_s[x(s), y(s)]$ of the smoothed trajectory, and a_x and a_y are the *x* and *y* components of the acceleration $\partial_s^2[x(s), y(s)]$ of the smoothed

*Because the sign of the curvature is important, and because we never observed curvatures greater than 1, we calculated histograms of (−sign(curvature) × log(|curvature|). The negative sign is necessary, because taking the logarithm reverses the sign on the curvature.

trajectory (in these expressions, s is a parameter indicating a point on the smoothed trajectory; as s increases from zero, the point under consideration moves from the beginning to the end of the trajectory). It is clear from this figure that neurites in the turning model tend to turn so as to increase their motion up the gradient. When their current bearing is to the left of the gradient (Fig. S6*C*), they tend to curve in a clockwise direction; when they are bearing to the right of the gradient (Fig. S6*E*), they tend to curve in an anticlockwise direction. In contrast, neurites in the growth-rate modulation model show no such corrective turning: Regardless of their direction of travel, they are equally likely to curve anticlockwise or clockwise. Furthermore, this behavior shows no obvious dependence on the magnitude of the gradient.

Proof That There Is No Average Curvature Toward the Gradient Under the Growth-Rate Modulation Model

In this section, we prove that there is no turning on average (as quantified by the curvature ratio, as defined in Fig. S2) in the growth-rate modulation model of axon guidance. We obtain this result by showing that, under the growth-rate modulation model, for every neurite trajectory that turns up the gradient there is another trajectory that is equally likely to occur but that turns down the gradient. Fig. S5*A* illustrates a pair of such trajectories. As a result of this symmetry, explants generated by the growth-rate modulation model have zero turning on average, because an explant is equally likely to exhibit positive turning (i.e., with a larger number of trajectories similar to path 1 in Fig. S5*A*) as negative turning (i.e., dominated by trajectories similar to path 2). Note that, as discussed in the previous section, both trajectories represent net progress up the gradient.

Definition. A path τ is defined by a sequence of angles θ_j and step lengths l_j:

$$\tau = (\theta_1, l_1; \theta_2, l_2; \ldots; \theta_n, l_n). \quad [4]$$

We write $\boldsymbol{\theta}_\tau$ for the sequence of angles and $\boldsymbol{l}_\tau$ for the sequence of step lengths, neglecting the subscript τ when it is obvious by context. The length of the path τ is $\sum_{j=1}^{n} l_j$.

Definition. The probability $P(\tau)$ of a path τ under the growth-rate modulation model is given by

$$\begin{aligned} P(\tau) &\equiv P(\boldsymbol{\theta}_\tau, \boldsymbol{l}_\tau) \\ &= P(\boldsymbol{l}_\tau|\boldsymbol{\theta}_\tau) P(\boldsymbol{\theta}_\tau), \end{aligned} \quad [5]$$

where $P(\boldsymbol{l}|\boldsymbol{\theta}) = \prod_{j=1}^{n} P(l_j|\theta_j)$ (i.e., each l_j depends only on the direction of the corresponding step) and

$$P(\boldsymbol{\theta}) = P_0\left(\theta_1\right) \prod_{j=2}^{n} P(\theta_j|\theta_{j-1}). \quad [6]$$

In this expression, $P_0(\theta_1)$ denotes the probability that the first step in the trajectory is in the direction θ_1. Because we assume that the gradient has no influence on the initial outgrowth direction of the neurites (see ref. 5 for justification), this probability is independent of θ_1, and we can write $P_0(\theta_1) = 1/2\pi$. $P(\theta_j|\theta_{j-1})$ gives the probability that the jth step in the path will be in the direction θ_j, given that the previous step was in the direction θ_{j-1}. Under the growth-rate modulation model, the growth cone is as likely to curve toward the gradient as away from it, and the magnitude of the turn also is independent of the gradient direction. Thus, $P(\theta_j|\theta_{j-1})$ is symmetric around $\theta_j = \theta_{j-1}$ and depends only on the difference between θ_j and θ_{j-1}:

$$P(\theta_j|\theta_{j-1}) = P(\theta_j - \theta_{j-1}|0) = P(\theta_{j-1} - \theta_j|0). \quad [7]$$

To measure the amount that a path turns in the direction of the gradient, we use the smoothing-spline approximation to the path $\boldsymbol{q}(s;\tau,\sigma)$ (where σ indicates the strength of the smoothing, and s indicates the distance along the smoothed path) obtained by trading off the accuracy to which the path is approximated against the smoothness of the approximating path (see details below). For each point s on this path, we use the directed curvature $\kappa[\boldsymbol{q}(r;\tau,\sigma)](s)$ of the path at that point to assign a value indicating whether the path is tending to curve toward or away from the gradient direction. (In this expression, r also represents a distance along the smoothed path; we use both r and s to distinguish between positions on the smoothed path and points at which the curvature is evaluated.) Informally, imagine the gradient is pointing upwards. Then, if the segment of the path under consideration is pointing to the right, a "clockwise" curvature would indicate turning away from the gradient, whereas an "anticlockwise" curvature would indicate turning toward the gradient, and vise-versa for a segment pointing to the left (illustrated in Fig. S5*B*). Formally, this situation is captured by the expression

$$\kappa\left[\boldsymbol{q}(r;\tau,\sigma)\right](s) \times \partial_s \boldsymbol{q}(s;\tau,\sigma) \cdot \hat{\theta}_{\text{gradient}}, \quad [8]$$

where $\partial_s \boldsymbol{q}(s;\tau,\sigma)$ is the "velocity" of the path at the point under consideration (∂_s represents partial differentiation with respect to s) and " $\times$ " indicates the vector cross product. If this expression is positive, the path is (locally) turning up the gradient; if the expression is negative, the path is turning down the gradient. The directed curvature $\kappa[\boldsymbol{q}(r;\tau,\sigma)](s)$ is a vector with magnitude related to the "sharpness" of a turn. Its direction is into the page for clockwise turns, and out of the page for anticlockwise turns (i.e., it is given by the right-hand rule). Thus, the cross-product of the velocity and the directed curvature gives the direction of the turn. Projecting this vector onto the direction of the gradient thus measures the rate of turning in the direction of the gradient, as expressed by Eq. **8**. We thus arrive at our formal definition of the turning measure for a path.

Definition. We define $\gamma_\sigma(\tau)$, the turning measure of the path τ, as the fraction (by length) of the smoothed path $\boldsymbol{q}(s;\tau,\sigma)$ that is turning in the direction of the gradient, minus the fraction of the smoothed path that is turning away from the direction of the gradient. Formally, we have

$$\begin{aligned} \gamma_\sigma(\tau) = &\frac{1}{\int_0^n |\partial_s \boldsymbol{q}(s;\tau,\sigma)| ds} \int_0^n \operatorname{sgn}\left(\kappa\left[\boldsymbol{q}(r;\tau,\sigma)\right](s)\right. \\ &\left. \times \partial_s \boldsymbol{q}(s;\tau,\sigma) \cdot \hat{\theta}_{\text{gradient}}\right) |\partial_s \boldsymbol{q}(s;\tau,\sigma)| ds. \end{aligned} \quad [9]$$

where $\boldsymbol{q}(r;\tau,\sigma)$, defined for s in $[0, n]$ is the doubly-differentiable function that minimizes the functional

$$H[\boldsymbol{q}(r),\tau] = \sum_{j=0}^{n} |\boldsymbol{r}_{\tau,j} - \boldsymbol{q}(j)|^2 + \sigma \int_0^n |\partial_s^2 \boldsymbol{q}(s)|^2 ds. \quad [10]$$

In this expression, $\boldsymbol{r}_{\tau,j}$ is defined by

$$\boldsymbol{r}_{\tau,j} = -\frac{1}{2} \sum_{l=1}^{n} \boldsymbol{v}_l + \sum_{k=1}^{j} \boldsymbol{v}_k. \quad [11]$$

where $\boldsymbol{v}_l = l_l \hat{\theta}_l$, with $\hat{\theta}$ a unit vector in the θ direction. The functional H trades off the smoothness of the resulting path against its closeness to the points it is meant to approximate. For example, as the path $\boldsymbol{q}(r)$ deviates from the points $\boldsymbol{r}_{\tau,j}$, the first term in H increases in magnitude, whereas a highly convoluted path will tend to increase the second term in H. σ controls the relative weighting of these two terms. Thus, when σ is large, the smoothness constraint becomes more important; when σ is small, the

path is influenced more by the target points. The directed curvature $\kappa[\boldsymbol{q}(r)](s)$ of the curve $\boldsymbol{q}(r)$ at s is defined by

$$\kappa[\boldsymbol{q}(r)](s) = \frac{\partial_r \boldsymbol{q}(r) \times \partial_r^2 \boldsymbol{q}(r)}{(\partial_r \boldsymbol{q}(r) \cdot \partial_r \boldsymbol{q}(r))^{3/2}}|_{r=s}. \quad [12]$$

Lemma 1. For each possible path τ, there exists another path τ' with the following properties:

1. $P(\tau') = P(\tau)$,
2. $l(\tau') = l(\tau)$ and
3. $\gamma_\sigma(\tau') = -\gamma_\sigma(\tau)$.

Proof. Given the path $\tau = (\theta_1, l_1; \ldots; \theta_n, l_n)$, we construct the path $\tau' = (\theta_n, l_n; \theta_{n-1}, l_{n-1}; \ldots; \theta_1, l_1)$. Then we have

$$\begin{aligned} P(\tau') &= (\textstyle\prod_{j=1}^{n} P(l_{n+1-j}|\theta_{n+1-j}))(P_0(\theta_n) \prod_{j=1}^{n-1} P(\theta_{n-j}|\theta_{n+1-j})) \\ &= P(\boldsymbol{l}_\tau|\boldsymbol{\theta}_\tau) P_0(\theta_n) \textstyle\prod_{j=1}^{n-1} P(\theta_{n+1-j}|\theta_{n-j}) \\ &= P(\boldsymbol{l}_\tau|\boldsymbol{\theta}_\tau) P_0(\theta_n) \textstyle\prod_{j=2}^{n} P(\theta_j|\theta_{j-1}) \\ &= P(\boldsymbol{l}_\tau|\boldsymbol{\theta}_\tau) \frac{P_0(\theta_n)}{P_0(\theta_1)} P(\boldsymbol{\theta}_\tau). \end{aligned} \quad [13]$$

Then, because $P_0(\theta_n) = P_0(\theta_1) = 1/2\pi$, we have

$$\begin{aligned} P(\tau') &= P(\boldsymbol{l}_\tau|\boldsymbol{\theta}_\tau) P(\boldsymbol{\theta}_\tau) \\ &= P(\tau), \end{aligned} \quad [14]$$

proving property 1. Property 2 follows trivially by noting that $l(\tau') = \sum_{j=1}^{n} l_{n+1-j} = \sum_{j=1}^{n} l_j = l(\tau)$.
It remains to show that $\gamma_\sigma(\tau') = -\gamma_\sigma(\tau)$. We first demonstrate that $\boldsymbol{q}(s;\tau',\sigma) = -\boldsymbol{q}(n-s;\tau,\sigma)$ by showing that $H[-\boldsymbol{q}(n-r),\tau'] = H[\boldsymbol{q}(r),\tau]$. We have

$$\begin{aligned} H[-\boldsymbol{q}(n-r),\tau'] = &\sum_{j=0}^{n} |\boldsymbol{r}_{\tau',j} - (-\boldsymbol{q}(n-j))|^2 \\ &+ \sigma \int_0^n |\partial_{n-s}^2(-\boldsymbol{q}(n-s))|^2 ds. \end{aligned} \quad [15]$$

We address this equation term-by-term. Note that

$$\begin{aligned} \boldsymbol{r}_{\tau',j} &= -\tfrac{1}{2}\textstyle\sum_{l=1}^{n} \boldsymbol{v}'_l + \sum_{k=1}^{j} \boldsymbol{v}'_k \\ &= -\tfrac{1}{2}\textstyle\sum_{l=1}^{n} \boldsymbol{v}_{n+1-l} + \sum_{k=1}^{j} \boldsymbol{v}_{n+1-k} \\ &= -\tfrac{1}{2}\textstyle\sum_{l=1}^{n} \boldsymbol{v}_l + \sum_{k=n+1-j}^{n} \boldsymbol{v}_k \\ &= -\tfrac{1}{2}\textstyle\sum_{l=1}^{n} \boldsymbol{v}_l + \sum_{l=1}^{n} \boldsymbol{v}_l + \sum_{k=1}^{n-j}(-\boldsymbol{v}_k) \\ &= -\tfrac{1}{2}\textstyle\sum_{l=1}^{n}(-\boldsymbol{v}_l) + \sum_{k=1}^{n-j}(-\boldsymbol{v}_k) \\ &= -\boldsymbol{r}_{\tau,n-j}. \end{aligned} \quad [16]$$

and thus the first term of Eq. **15** is

$$\begin{aligned} \text{term 1} &= \textstyle\sum_{j=0}^{n} |-\boldsymbol{r}_{\tau,n-j} - (-\boldsymbol{q}(n-j))|^2 \\ &= \sum_{j=0}^{n} |\boldsymbol{r}_{\tau,j} - \boldsymbol{q}(j)|^2. \end{aligned} \quad [17]$$

For the second term, we have (changing variables from s to $v = n-s$)

$$\begin{aligned} \text{term 2} &= \sigma \textstyle\int_0^n |\partial_{n-s}^2(-\boldsymbol{q}(n-s))|^2 ds \\ &= -\sigma \int_n^0 |\partial_v^2 \boldsymbol{q}(v)|^2 dv \\ &= \sigma \textstyle\int_0^n |\partial_v^2 \boldsymbol{q}(v)|^2 dv. \end{aligned} \quad [18]$$

Combining these expressions, we see that, indeed, $H[-\boldsymbol{q}(n-r),\tau'] = H[\boldsymbol{q}(r),\tau]$. This equality in turn implies that $\boldsymbol{q}(s;\tau',\sigma) = -\boldsymbol{q}(n-s;\tau,\sigma)$, because $\boldsymbol{q}(s;\tau,\sigma)$ is the path that minimizes $H[\boldsymbol{q}(r),\tau]$, and thus $-\boldsymbol{q}(n-s;\tau,\sigma)$ is the path that minimizes $H[\boldsymbol{q}(r),\tau']$.

It is now straightforward to show that $\gamma_\sigma(\tau') = -\gamma_\sigma(\tau)$. From the definition of $\gamma_\sigma(\tau')$, we have

$$\begin{aligned} \gamma_\sigma(\tau') &= \frac{1}{l}\int_0^n \operatorname{sgn}\big(\kappa[\boldsymbol{q}(r;\tau',\sigma)](s) \\ &\quad \times \partial_s \boldsymbol{q}(s;\tau',\sigma)\cdot\widehat{\theta}_{\text{gradient}}\big)\,|\partial_s \boldsymbol{q}(s;\tau',\sigma)|ds \\ &= \frac{1}{l}\int_0^n \operatorname{sgn}(\kappa[-\boldsymbol{q}(n-r;\tau,\sigma)](s) \\ &\quad \times -\partial_s \boldsymbol{q}(n-s;\tau,\sigma)\cdot\widehat{\theta}_{\text{gradient}}\big)\,|\partial_s \boldsymbol{q}(n-s;\tau,\sigma)|ds \\ &= \frac{1}{l}\int_0^n \operatorname{sgn}\big(\kappa[-\boldsymbol{q}(n-r;\tau,\sigma)](s) \\ &\quad \times \partial_{n-s} \boldsymbol{q}(n-s;\tau,\sigma)\cdot\widehat{\theta}_{\text{gradient}}\big)\,|\partial_{n-s} \boldsymbol{q}(n-s;\tau,\sigma)|ds, \end{aligned} \quad [19]$$

where $l = \int_0^n |\partial_s \boldsymbol{q}(s;\tau',\sigma)|ds$ is the length of $\boldsymbol{q}(s;\tau',\sigma)$. The curvature $\kappa[-\boldsymbol{q}(n-r;\tau,\sigma)](s)$ can be rewritten

$$\begin{aligned} \kappa[-\boldsymbol{q}(n-r;\tau,\sigma)](s) &= \frac{\partial_r(-\boldsymbol{q}(n-r;\tau,\sigma)) \times \partial_r^2(-\boldsymbol{q}(n-r;\tau,\sigma))}{(\partial_r(-\boldsymbol{q}(n-r;\tau,\sigma))\partial_r(-\boldsymbol{q}(n-r;\tau,\sigma)))^{3/2}}|_{r=s} \\ &= \frac{\partial_r(\boldsymbol{q}(n-r;\tau,\sigma)) \times \partial_r^2(\boldsymbol{q}(n-r;\tau,\sigma))}{(\partial_r(\boldsymbol{q}(n-r;\tau,\sigma))\partial_r(\boldsymbol{q}(n-r;\tau,\sigma)))^{3/2}}|_{r=s} \\ &= \frac{(-\partial_v \boldsymbol{q}(v;\tau,\sigma)) \times \partial_v^2 \boldsymbol{q}(v;\tau,\sigma)}{(\partial_v \boldsymbol{q}(v;\tau,\sigma)\partial_v \boldsymbol{q}(v;\tau,\sigma))^{3/2}}|_{v=n-s} \\ &= -\kappa[\boldsymbol{q}(r;\tau,\sigma)](n-s). \end{aligned} \quad [20]$$

and thus, substituting Eq. **20** and $v = n-s$ into Eq. **19**, we finally obtain

$$\begin{aligned} \gamma_\sigma(\tau') &= \frac{1}{l}\int_0^n \operatorname{sgn}(-\kappa[\boldsymbol{q}(r;\tau,\sigma)](n-s) \\ &\quad \times \partial_{n-s} \boldsymbol{q}(n-s;\tau,\sigma)\widehat{\theta}_{\text{gradient}})\,|\partial_{n-s} \boldsymbol{q}(n-s;\tau,\sigma)|ds \\ &= -\frac{1}{l}\int_n^0 \operatorname{sgn}(-\kappa[\boldsymbol{q}(r;\tau,\sigma)](v) \\ &\quad \times \partial_v \boldsymbol{q}(v;\tau,\sigma)\widehat{\theta}_{\text{gradient}})\,|\partial_v \boldsymbol{q}(v;\tau,\sigma)|;dv \\ &= -\frac{1}{l}\int_0^n \operatorname{sgn}(\kappa[\boldsymbol{q}(r;\tau,\sigma)](v) \\ &\quad \times \partial_v \boldsymbol{q}(v;\tau,\sigma)\widehat{\theta}_{\text{gradient}})\,|\partial_v \boldsymbol{q}(v;\tau,\sigma)|dv = -\gamma_\sigma(\tau), \end{aligned} \quad [21]$$

proving property 3, and completing the proof of lemma 1.

In our model, an explant X is completely characterized by a set of independently generated neurite trajectories $\Omega_X = \{\tau_i\}$. We define the turning measure for an explant in terms of the turning measures of its constituent trajectories by

$$\Gamma_\sigma(X) = \frac{\sum_{\tau\in\Omega_X} l(\tau)\gamma_\sigma(\tau)}{\sum_{\tau\in\Omega_X} l(\tau)}. \quad [22]$$

$\Gamma_\sigma(X)$ is thus the length-weighted average of the turning measures of each of the trajectories in X.

Theorem 1. Under the growth-rate modulation model, the expected turning averaged over all explants is zero, $\Gamma_\sigma(X) = 0$, regardless of the smoothing level σ.

Proof. The proof follows directly from lemma 1 and the definition of $\Gamma_\sigma(X)$

$$\begin{aligned}\langle\Gamma_\sigma(X)\rangle &= \sum_X \Gamma_\sigma(X)P(X)\\ &= \sum_\Omega \frac{\sum_{\tau\in\Omega} l(\tau)\gamma_\sigma(\tau)}{\sum_{\tau\in\Omega} l(\tau)} \prod_{\tau\in\Omega} P(\tau)\\ &= \frac{1}{2}\sum_\Omega \left(\frac{\sum_{\tau\in\Omega} l(\tau)\gamma_\sigma(\tau)}{\sum_{\tau\in\Omega} l(\tau)} \prod_{\tau\in\Omega} P(\tau) + \frac{\sum_{\tau\in\Omega} l(\tau')\gamma_\sigma(\tau')}{\sum_{\tau\in\Omega} l(\tau')} \prod_{\tau\in\Omega} P(\tau')\right)\\ &= \frac{1}{2}\sum_\Omega \left(\frac{\sum_{\tau\in\Omega} l(\tau)\gamma_\sigma(\tau)}{\sum_{\tau\in\Omega} l(\tau)} - \frac{\sum_{\tau\in\Omega} l(\tau)\gamma_\sigma(\tau)}{\sum_{\tau\in\Omega} l(\tau)}\right) \prod_{\tau\in\Omega} P(\tau)\\ &= 0.\end{aligned} \quad [23]$$

1. Freeman WT, Adelson EH (1991) The design and use of steerable filters. *IEEE Trans Pattern Anal Mach Intell* 13:891–906.
2. Meijering E, Jacob M, Sarria J-CF, Unser M (2003) A novel approach to neurite tracing in fluorescent microscopy images. *Proceedings of the Fifth IASTED International Conerence on Signal and Image Processing*, ed, Hamza MH (ACTA press, Calgary, AB, Canada) p 491.
3. Meijering E, et al. (2004) Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytometry A* 58:167–176.
4. Meijering E, et al. (2004) Neurite tracing in fluorescence microscopy images using ridge filtering and graph searching: Principles and validation. *Proceedings of the IEEE International Symposium on Biomedical Imaging: Nano to Macro*, Vol 2, pp 1219–1222.
5. Mortimer D, et al. (2009) Bayesian model predicts the response of axons to molecular gradients. *Proc Natl Acad Sci USA* 106:10296–10301.

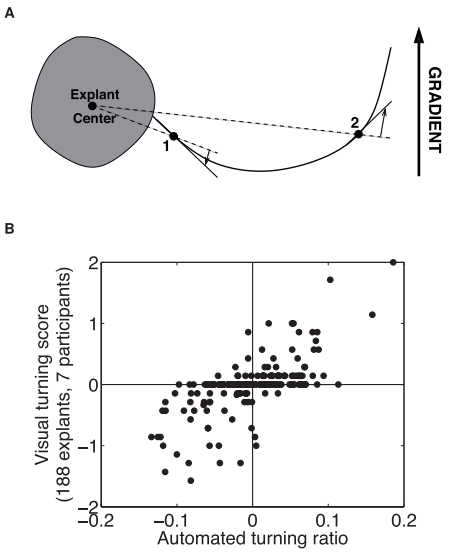

Fig. S1. The turning ratio guidance measure. (*A*) Neurite pixels are classified as turning up or down the gradient, based on their orientation (as calculated by the ridge-filter) and their position relative to the center of the explant. In this example, pixel 1 is regarded as turning down the gradient, because the component of its orientation orthogonal to the radial vector extending from the explant center is pointing away from the gradient direction. In contrast, pixel 2 is regarded as turning up the gradient. In this schematic, for simplicity, a complete trajectory is shown. However, because of the high density of neurites in our images, it was not possible to reconstruct complete trajectories reliably, so an analysis of turning in terms of the behavior of neighboring trajectory segments (e.g., curvature) was not possible. (*B*) Scatterplot showing correlation between human-scored and automatically determined measures of turning. Pearson coefficient of correlation = 0.66. For the four explants shown in Fig. 1*A* of the main text, the mean human-scored values were 0, 0, 0.14, and 2, respectively.

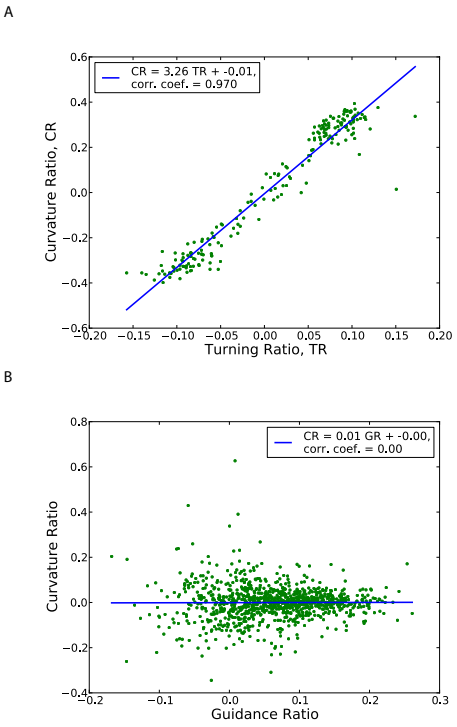

Fig. S2. (*A*) Turning ratio correlates strongly with curvature ratio for explants generated using an explicit turning mechanism. (*B*) There is no correlation between guidance ratio (GR) and curvature ratio for explants generated through a growth-rate modulation mechanism, suggesting that guidance in these explants occurs without turning. The curvature ratio for an explant was calculated by smoothing each neurite trajectory on a length scale comparable to the neurite length using a smoothing spline (the MATLAB spaps function), calculating the curvature of the smoothed trajectory at each point, and counting the number of points curving up the gradient (n_u) and the number of points curving down the gradient (n_d). These numbers were summed over all neurite trajectories to obtain N_u and N_d. The curvature ratio then was calculated as $(N_u-N_d)/(N_u+N_d)$.

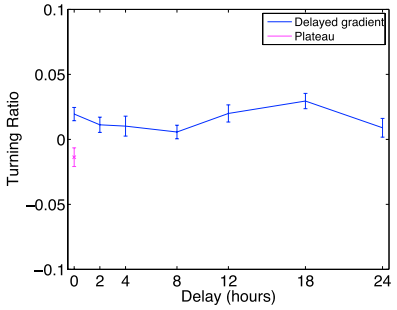

Fig. S3. Delaying the application of the gradient does not produce strong turning. Although for all delays there was significant turning compared with the zero-gradient condition (for *n* and *P* values, see Table S2), in all cases the turning was weak and did not depend on the time of gradient application.

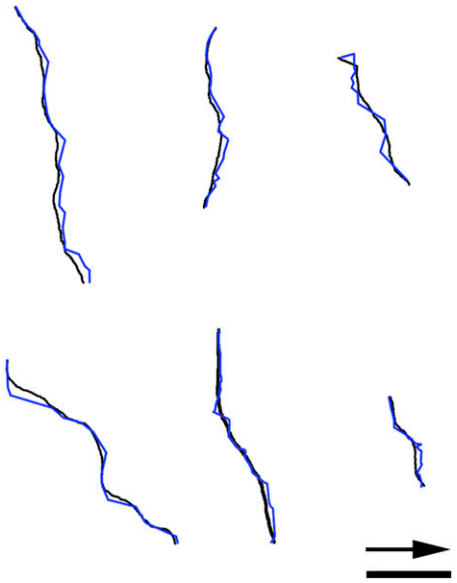

Fig. S4. Neurite trajectories do not straighten over time. Neurite traces after growth in an NGF gradient in 3D collagen. The position of a growth cone during time-lapse imaging (blue line) is very similar to the final trace of the neurite (black line), indicating that neurites in collagen do not straighten substantially (mean length of black lines is 5.6 ± 1.5% shorter than blue lines). Growth in all panels is from bottom to top. Arrow shows direction of gradient. (Scale bar, 50 μm.)

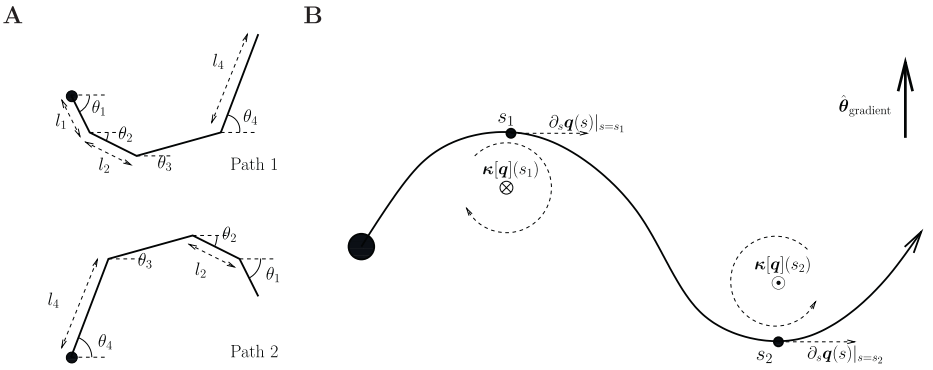

Fig. S5. Details of curvature-based turning measure. (*A*) Two paths with equal probabilities under the growth-rate modulation model but with opposite turning. Path 2 is obtained from Path 1 by reversing the order in which each step is taken. We prove below that these two paths have equal probability under the growth-rate modulation model. (*B*) Curvature-based turning quantification on a smoothed path. The path $\mathbf{q}(s)$ has opposite turning at the two points, s_1 and s_2. At s_1, $\boldsymbol{\kappa}[\mathbf{q}](s_1) \times \partial_s \mathbf{q}(s)|_{s=s1}$ points in the direction opposite the direction of the gradient $\theta_{gradient}$, consistent with the neurite turning away from the gradient at that point. The reverse is true at s_2.

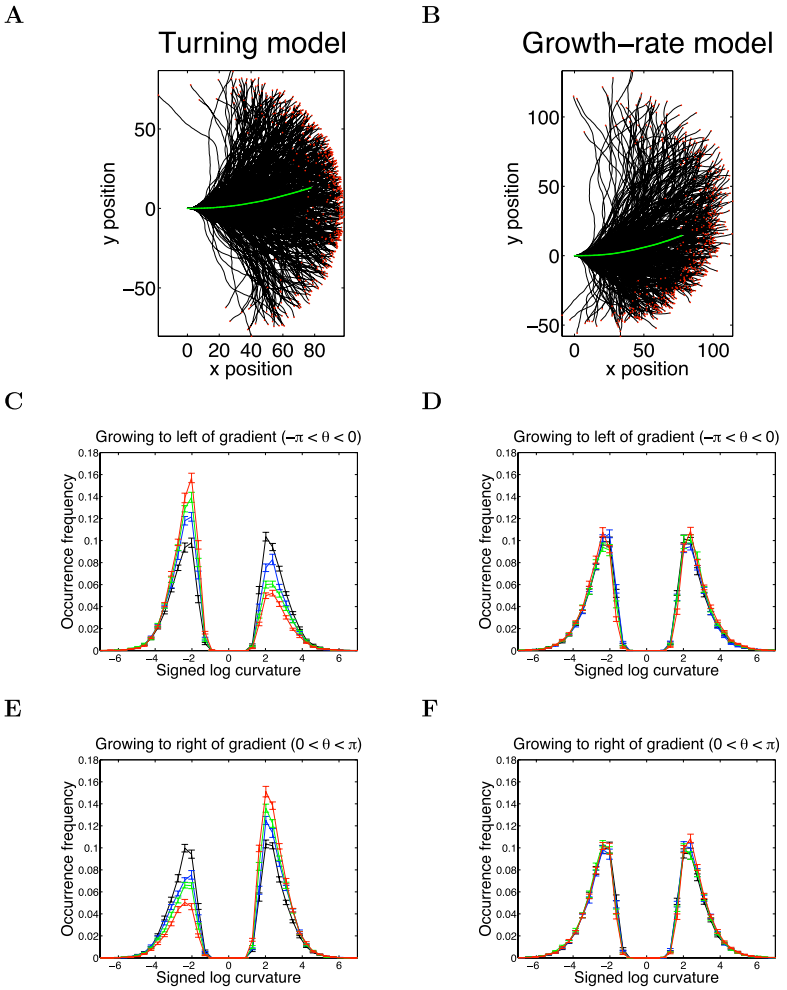

Fig. S6. Turning in individual and average trajectories. (*A* and *B*) Trajectories generated by the growth-rate modulation model (*A*) and turning model (*B*) when initially all neurites are directed along the positive *x* axis (gradient pointing up). Note that in both cases the average trajectory (in green) turns toward the gradient. (*C*–*F*) Normalized conditional histograms of the ``signed logarithm'' of the curvature for the turning model (*C* and *E*) and the growth-rate modulation model (*D* and *F*) over a range of gradient strengths. Each histogram combines all points on each neurite trajectory for which the neurite's current growth direction θ is to the left (*C* and *D*; $-\pi < \theta < 0$) or right (*E* and *F*; $0 < \theta < \pi$) relative to the gradient. Positive (negative) signed log curvature corresponds to an anticlockwise (clockwise) curving trajectory. Thus, panels *C* and *E* show that under the turning model, neurites growing to the left (or right) of the gradient tend to correct by turning clockwise (or anticlockwise) and that the tendency of neurites to correct their direction is enhanced by increasing the gradient strength. In contrast, no such correction occurs in the growth-rate modulation model (*D* and *F*). Black: $\delta_T = 0$, $\delta_{GRM} = 0$; blue: $\delta_T = 0.05$, $\delta_{GRM} = 0.5$; green: $\delta_T = 0.1$, $\delta_{GRM} = 1$; red: $\delta_T = 0.15$, $\delta_{GRM} = 1.5$.

Table S1. Turning ratio (TR) values for the 0.12%, 0.18%, 0.24%, and 0.30% gradients at each NGF concentration, tabulated with the *P* values compared with the plateau-condition sample

0.12% gradient

[NGF]	0.00092 nM	0.0092 nM	0.028 nM	0.092 nM	0.28 nM
N	29	63	66	80	88
p	0.2	0.9	0.7	0.02	0.0006
TR	0.003	−0.011	−0.016	0.005	0.012
[NGF]	0.92 nM	2.8 nM	9.2 nM	28 nM	92 nM
N	108	46	38	14	15
p	0.01	0.6	0.5	0.5	0.7
TR	0.004	−0.007	−0.022	0.000	−0.005

0.18% gradient

[NGF]		0.011 nM	0.034 nM	0.11 nM	0.34 nM
N		64	71	75	74
p		0.002	0.2	0.004	0.1
TR		0.014	−0.002	0.011	0.000
[NGF]	1.1 nM	3.4 nM	11 nM	34 nM	110 nM
N	113	49	45	50	19
p	0.01	0.5	0.02	0.2	0.5
TR	0.004	−0.006	0.012	0.001	−0.004

0.24% gradient

[NGF]	0.0015 nM	0.015 nM	0.045 nM	0.15 nM	0.45 nM
N	47	71	90	91	94
p	0.7	0.8	0.4	0.7	0.001
TR	−0.009	−0.011	−0.019	−0.010	0.011
[NGF]	1.5 nM	4.5 nM	15 nM	45 nM	
N	112	74	63	36	
p	0.03	0.1	0.03	0.01	
TR	0.002	−0.013	−0.034	−0.045	

0.30% gradient

[NGF]	0.0021 nM	0.021 nM	0.064 nM	0.21 nM	0.64 nM
N	54	67	69	68	72
p	0.8	0.09	0.3	0.02	0.0007
TR	−0.010	0.001	−0.004	0.007	0.016
[NGF]	2.1 nM	6.4 nM	21 nM	64 nM	
N	105	82	92	68	
p	0.1	0.002	0.5	0.4	
TR	−0.001	−0.041	−0.006	−0.004	

Table S2. Sample sizes and turning ratios for each condition in the delayed application experiment, shown with *P* values compared with the plateau-condition sample

Delay (h)	*N*	*P*	Turning ratio
0	124	0.0003	0.019
2	85	0.01	0.011
4	59	0.03	0.010
8	60	0.04	0.0057
12	59	0.001	0.020
18	96	10^{-5}	0.029
24	88	0.04	0.0089
Plateau	59		−0.014