APPENDIX

In the results section we provided the descriptions of the new image analysis program. Here we present the mathematical details.

A.1 Tracking marker point on the root images

(a) *Highest correlation coefficient search*

After the user selects the points on the reference frame and the points are interpolated, the user specifies a search box size of R pixels within which the new location of a tracked point is searched in the current frame. The user also specifies a template size N which is used to calculate the correlation coefficient between the reference image and the current image. For example, if the coordinate of a selected point in the reference image (Fig. 2A) is (x_0, y_0), then the new location (x^*, y^*) of this point in the current image (Fig. 2B) is searched within an $R \times R$ search box surrounding the point (x_0, y_0). Let the intensity of pixels within an $N \times N$ box surrounding the point (x_0, y_0) in the reference image be $I_0(x, y)$, and in the current image the intensity of $N \times N$ box of pixels centered around the point (x^*, y^*) be $I^*(x, y)$. The correlation coefficient between these two boxes of pixels is

$$C = \frac{\sum_{x=-\frac{N}{2}}^{\frac{N}{2}} \sum_{y=-\frac{N}{2}}^{\frac{N}{2}} (I_0(x+x_0, y+y_0) - \bar{I}_0)(I^*(x+x^*, y+y^*) - \bar{I}^*)}{\sqrt{\sum_{x=x_0-\frac{N}{2}}^{x_0+\frac{N}{2}} \sum_{y=y_0-\frac{N}{2}}^{y_0+\frac{N}{2}} (I_o(x,y) - \bar{I}_o)^2 \sum_{x=x^*-\frac{N}{2}}^{x^*+\frac{N}{2}} \sum_{y=y^*-\frac{N}{2}}^{y^*+\frac{N}{2}} (I^*(x,y) - \bar{I}^*)^2}}, \quad (1)$$

where $\bar{I}_0$ and $\bar{I}^*$ are average pixel intensities i.e.

$$\bar{I}_0 = \frac{1}{N^2} \sum_{x=x_0-\frac{N}{2}}^{x_0+\frac{N}{2}} \sum_{y=y_0+\frac{N}{2}}^{y_0+\frac{N}{2}} I_0(x, y) \quad (2)$$

$$\bar{I}^* = \frac{1}{N^2} \sum_{x=x^*-\frac{N}{2}}^{x^*+\frac{N}{2}} \sum_{y=y^*-\frac{N}{2}}^{y^*+\frac{N}{2}} I^*(x, y). \quad (3)$$

For a point located at (x_0, y_0) in the reference frame, the correlation coefficient C is calculated for all points (x^*, y^*) located within $x_0 - \frac{R}{2} \le x^* \le x_0 + \frac{R}{2}$ and $y_0 - \frac{R}{2} \le y^* \le y_0 + \frac{R}{2}$ in the current frame. The point where correlation coefficient C is highest is chosen as the new location (x^*, y^*) of the point (x_0, y_0) in the current image. In the rare case where the maximal correlation point is not unique, the average coordinate of all points where correlation coefficient C is highest is chosen as the new location (x^*, y^*).

From a separate study of overall root growth we found that the maximum expected displacement $\Delta x \approx 0.1$ mm of any marker point over five minutes. Therefore an optimal value for the size of search box $R = 2\Delta x = 0.2$ mm was used to ensure that even in the rare case of very high growth rate the algorithm could track the marker points while avoiding unnecessary computations. For estimating the optimum value of N, the average spacing of patterns on the last image in the time sequence was used because as the root grows the gap between the graphite particles increases in the growth zone and in the last image the gap between two graphite particles becomes highest. In our measurements N = 0.25 mm produced the most optimum results.

Since all of the images were colored, we calculated the correlation coefficients C_r, C_g and C_b for red, green and blue color intensities respectively using equation (1) where I_0 and I^* were replaced by the appropriate color intensities. For searching the highest correlation coefficient, the user could choose the average of C_r, C_g and C_b as an estimate of the overall similarity between two boxes or any of the C_r, C_g and C_b values.

In order to make the algorithm efficient, the user had the choice of using the velocity of the individual points to provide a better initial guess to the search algorithm. When a point (x_0, y_0) moves to (x^*, y^*) in time δt the two-dimensional velocity of the point is

$$u = \frac{x^* - x_0}{\delta t}, v = \frac{y^* - y_0}{\delta t}. \tag{4}$$

Therefore to find the location (x', y') in the following image we could limit our search area within $x^* + u\delta t - \frac{R}{2} \leq x' \leq x^* + u\delta t + \frac{R}{2}$ and $y^* + v\delta t - \frac{R}{2} \leq y' \leq y^* + v\delta t + \frac{R}{2}$ under the assumption that the change in velocity of a point is small within a span of three consecutive images. This approach of using the velocity of the points to predict the new location of the points helps the program track all points more accurately and efficiently.

(b) *Highest color weighted correlation coefficient search algorithm*

The highest correlation search method described above matches boxes of pixels irrespective of whether the pixels are on the root or on the germination paper behind it. Although this method works in more than 70% of the kinematic studies, in the cases where as the root grows and moves into an area of the germination paper where the texture of the paper is very different from the reference image, the algorithm tends to struggle to track the points accurately.

To overcome this problem we improved the algorithm by introducing a weighing factor $w(x, y)$ based on color of the pixel into the calculation of correlation coefficient *C*. With this new algorithm the user selects a small area of the image covering only the root and then another area covering only the germination paper. Colors from each of these areas are averaged and stored as root color (R_r, G_r, B) and background color (R_b, G_b, B_b) where *R*, *G* and *B* are the intensities of red, green and blue respectively, and range between 0 and 1. Figure 6 shows a schematic for calculating the weighing factor *w*. If the difference of intensity of any color between the root and the background is less than 0.2, the weighing factor *w* is assigned a value of 1 (e.g. the blue color in Fig. 3), otherwise *w* is calculated by linear interpolation for pixels which has color intensity between that of the root and the background. If the color intensity is outside the root-background color intensity range, *w* is assigned a value of 1 or 0 depending on proximity to the root color or background color respectively. Using the weighing factor the color weighted correlation coefficient is defined as

$$C_c = \frac{\sum_{x=-\frac{N}{2}}^{\frac{N}{2}} \sum_{y=-\frac{N}{2}}^{\frac{N}{2}} (I_0(x+x_0, y+y_0) - \bar{I}_0)(I^*(x+x^*, y+y^*) - \bar{I}^*) w_0(x+x_0, y+y_0) w^*(x+x^*, y+y^*)}{W \sqrt{\sum_{x=x_0-\frac{N}{2}}^{x_0+\frac{N}{2}} \sum_{y=y_0-\frac{N}{2}}^{y_0+\frac{N}{2}} (I_o(x,y) - \bar{I}_o)^2 \sum_{x=x^*-\frac{N}{2}}^{x^*+\frac{N}{2}} \sum_{y=y^*-\frac{N}{2}}^{y^*+\frac{N}{2}} (I^*(x,y) - \bar{I}^*)^2}} \quad (5)$$

where w_0 and w^* are the weighing factors for pixels in the reference image and the current image respectively, and $W = \frac{1}{N^2} \sum_{x=-\frac{N}{2}}^{\frac{N}{2}} \sum_{y=-\frac{N}{2}}^{\frac{N}{2}} w_0(x+x_0, y+y_0) w^*(x+x^*, y+y^*)$.

The color-based weighing factors reduce the importance of the pixels from the background paper in calculating the correlation coefficients between two boxes of pixels. As a result, even if the texture of the background paper changes drastically the software is able to track the points on the root reliably. It should be noted that in case of low contrast images where the intensity difference between the root and background is less than 0.2 in all three colors the weighing factor becomes 1. As a result the color weighted highest correlation search approach changes to highest correlation search approach described in the previous section.

(c) *Minimum pixel intensity difference search algorithm*: Although the maximal correlation search method track points reliably, calculation of correlation coefficients for each point is computationally expensive and relatively slow. Therefore, for roots that do not bend, an easier approach is to calculate the normalized root-mean-square difference in color intensity between two boxes of pixels

$$C_D = \frac{\sqrt{\frac{1}{N^2}\sum_{x=-\frac{N}{2}}^{\frac{N}{2}}\sum_{y=-\frac{N}{2}}^{\frac{N}{2}}[I_0(x_0+x, y_0+y) - I^*(x^*+x, y^*+y)]^2}}{\frac{1}{N^2}\sum_{x=-\frac{N}{2}}^{\frac{N}{2}}\sum_{y=-\frac{N}{2}}^{\frac{N}{2}} I_0(x_0+x, y_0+y)}, \qquad (6)$$

and find the point where C_D between the two boxes is minimum. This method cannot be used for curving roots, since the relative movement of the graphite particles within a box of pixels is two-dimensional and highly inhomogeneous, i.e. when the intensity difference between one subgroup of pixels is minimal, the intensity difference in another subgroup becomes large. As a result the software cannot determine the absolute minimum value of C_D and this tracking method becomes inconsistent. This problem does not arise when the root grows along a straight line, which minimizes the relative movement of graphite particles within a particular box of pixels. For the minimum pixel intensity difference search method, the confidence measure is given by $F = 1 - C_{\mathrm{Dmax}}$ where $C_{D\mathrm{max}}$ is the maximum of the lowest normalized root-mean-square differences in color intensity between two boxes of pixels for tracking all marker points in all frames.

A.2 : Automatic Edge Detection and Finding the Midline of the Roots

(a) Noise smoothing and image gradient— Before detecting the edge of the root we smooth noise by convolving the image with a Gaussian filter. A two-dimensional Gaussian filter is defined as

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)}, \qquad (7)$$

and is shown in Figure 4A. The parameter σ controls the spread of the filter contributing to the blurriness of the filtered image which can be changed in KineRoot by changing filter radius. The pixel intensity $I(x, y)$ is convolved with this filter for noise smoothing. Figure 7B shows a close up image of a basal root and Figure 7C shows the convolved image

$$\begin{aligned}\tilde{I}(x,y) &= \int_{\xi}\int_{\eta} I(x-\xi, y-\eta)G(\xi,\eta)d\eta d\xi \\ &= \sum_{i}\sum_{j} I(x-i, y-j)G(i,j).\end{aligned} \tag{8}$$

As a result of convolution with the Gaussian filter, the image gets blurred which allows better calculation of the color gradient of the image. Figure 4D shows the magnitude of the color gradient $\left(\sqrt{\left(\frac{\partial \tilde{I}}{\partial x}\right)^2 + \left(\frac{\partial \tilde{I}}{\partial y}\right)^2}\right)$ of the image averaged for red, green and blue colors. The darker streaks in the image which represent higher color gradient show the edges in the image.

(b) Algorithm for identification of root midline— The midline of the root is identified by an algorithm which ensures that the midline is detected accurately irrespective of the location of the tracked points on the root and the curvature of the root. The flow chart in Figure 13A illustrates the steps of the algorithm for identification of the root midline. Figure 13B shows an example image with tracked points, root edge points and root midline identified. Although the algorithm identifies the root midline accurately without the need for the iterative loop (steps 2-3-4-5) when the two edges of the root are nearly parallel, it may fail when the edges are not parallel, e.g. near the root tip. The iterative algorithm is developed to overcome this problem resulting in accurate identification of the root midline in all possible cases.