RootNav: Navigating Images of Complex Root Architectures[1][C][W]

Michael P. Pound, Andrew P. French, Jonathan A. Atkinson, Darren M. Wells, Malcolm J. Bennett, and Tony Pridmore*

Centre for Plant Integrative Biology, School of Biosciences, University of Nottingham, Sutton Bonington LE12 5RD, United Kingdom (M.P.P., A.P.F., J.A.A., D.M.W., M.J.B., T.P.); and School of Computer Science, University of Nottingham, Jubilee Campus, Nottingham NG8 1BB, United Kingdom (A.P.F., T.P.)

We present a novel image analysis tool that allows the semiautomated quantification of complex root system architectures in a range of plant species grown and imaged in a variety of ways. The automatic component of RootNav takes a top-down approach, utilizing the powerful expectation maximization classification algorithm to examine regions of the input image, calculating the likelihood that given pixels correspond to roots. This information is used as the basis for an optimization approach to root detection and quantification, which effectively fits a root model to the image data. The resulting user experience is akin to defining routes on a motorist's satellite navigation system: RootNav makes an initial optimized estimate of paths from the seed point to root apices, and the user is able to easily and intuitively refine the results using a visual approach. The proposed method is evaluated on winter wheat (*Triticum aestivum*) images (and demonstrated on Arabidopsis [*Arabidopsis thaliana*], *Brassica napus*, and rice [*Oryza sativa*]), and results are compared with manual analysis. Four exemplar traits are calculated and show clear illustrative differences between some of the wheat accessions. RootNav, however, provides the structural information needed to support extraction of a wider variety of biologically relevant measures. A separate viewer tool is provided to recover a rich set of architectural traits from RootNav's core representation.

The root system plays a crucial role in the plant's acquisition of resources from the soil. Because the majority of real-world locations are resource limited, the method a plant adopts to explore the soil space is key to efficient water and nutrient uptake. Even where resources are less scarce, as in heavily supplemented commercial farming environments, optimizing root architecture can lead to more efficient use of water and nutrients and hence less need for irrigation and fertilization (Lynch, 1995). Linking the variations in architecture to genetic origins will be central to this optimization approach, yet is complicated by the differences in architectural appearance due to environmental conditions (de Dorlodot et al., 2007).

To begin to understand root system architectures (RSAs), first it is necessary to image the roots themselves, then to extract architectural information. While fast and effective manual methods of phenotyping have been developed that do not even require imaging (Trachsel et al., 2011), the resolution of digital imaging combined with the objectivity of automated analysis allows for more repeatable extraction of a much richer set of measurements.

Imaging plants growing on plates on clear gel is a common technique (French et al., 2009; Naeem et al., 2011), ideal from an imaging standpoint, as the roots are easily visible. Alternatives include growing roots on paper, rolled to save space and reduce light (Hetz et al., 1996), then unrolled to analyze the root system. Flatbed scanners can be used to acquire images of seedling roots growing along an agar surface (Bengough et al., 2004), aeroponically grown systems (Le Bot et al., 2010; Lobet et al., 2011), and rhizotrons (Gasch et al., 2011).

Once captured, the images must be analyzed in an efficient and objective manner, and to that end, a number of specific software tools have been developed to handle different imaging modalities and research requirements. Most rely on low-level, 'bottom-up' approaches, which apply a sequence of filters to successively remove pixels considered unlikely to correspond to root material, finally grouping those remaining together as roots. Common techniques used to achieve this are noise reduction (image smoothing), thresholding (selecting pixels above a given brightness), and skeletonization (thinning selected groups of pixels down to lines one pixel in diameter). The key drawback of the bottom-up approach is that errors accumulate: an incorrect decision made by one process commonly introduces further errors later

[1] This work was supported by Biotechnology and Biological Sciences Research Council and Engineering and Physical Sciences Research Council Centre for Integrative Systems Biology program funding to the Centre for Plant Integrative Biology and Biotechnology and Biological Sciences Research Council Professorial Fellowship funding (to J.A.A., D.M.W., and M.J.B.).

* Corresponding author; e-mail tony.pridmore@nottingham.ac.uk.

The author responsible for distribution of materials integral to the findings presented in this article in accordance with the policy described in the Instructions for Authors (www.plantphysiol.org) is: Tony Pridmore (tony.pridmore@nottingham.ac.uk).

[C] Some figures in this article are displayed in color online but in black and white in the print edition.

[W] The online version of this article contains Web-only data.

www.plantphysiol.org/cgi/doi/10.1104/pp.113.221531

in the pipeline. In the worst cases, this can initiate a cascading effect that can seriously affect final results. Poor noise reduction can lead to thresholding errors, for example. Imperfect thresholding in turn disrupts skeletonization, adding spurious segments and introducing breaks into true roots. One response to this problem is to quantify only broader root measures, such as convex hulls or overall root depth (Galkovskyi et al., 2012), which do not require detailed descriptions of the root architecture (nested structure, emergence angles, etc.). The alternative is to provide interactive tools (Armengaud, 2009; Clark et al., 2011) that allow the user to clean up and correct automatically generated results. This can be hard to achieve satisfactorily when large corrections are needed. More importantly, though most users are willing to spend some time interacting with their data to refine results, manual correction of obvious errors can be frustrating and requires the user to carefully scrutinize all of the automatically reported results. It tends to reduce both the confidence in, and use of, tools that rely upon it excessively.

By contrast, top-down approaches match a clearly stated model of the object(s) of interest to the input data. Models may be explicit topological or geometrical structures, such as the network of contours employed by CellSet (Pound et al., 2012) or a loose collection of assumptions embedded in a search process (Pridmore et al., 2012). Though requiring a suitable model to be identified a priori, top-down methods can be more resistant to global image clutter and intensity change (French et al., 2009). The top-down approach focuses attention on the properties of target objects, roots, rather than on the properties of images of roots. As a result, top-down methods are frequently applicable, or at least more easily adapted, to a wider range of data sets.

While fully automatic methods remain the goal, in many cases, the complexity of root architectures and the poor quality of their images make user interaction a necessity. Unsupported, manual extraction of RSA data (Le Bot et al., 2010) becomes impractical as experimental size increases, but semiautomatic image analysis methods, which comprise automatic and interactive components, can bring advantages (Lobet et al., 2011). Some experimental designs may require data to be gathered from only a subset of the visible roots, for example those affected by particular local conditions, which must be selected by the user. RSAscan contain many root branches, with image quality varying across the root system. It may be better to adopt an interactive sampling strategy, gathering reliable data from selected roots rather than potentially unreliable measurements, automatically, from the whole plant (Lobet et al., 2011).

User interaction with bottom-up image analysis methods typically focuses on error correction. Interaction with a top-down automatic component is more likely to involve manipulation of object (i.e. root) descriptions than the detection and removal of spurious or addition of missing, and often small, image regions and features. Users are involved in fitting the tool's model to the input image, a task more closely related to their own goals, and which makes better use of their knowledge of the application domain. Efficient selection of areas of interest is also more naturally achieved when the user can specify which roots, rather than which areas of the image, should be processed. Overall, interaction with a top-down system is typically much less onerous.

Here, we describe RootNav, a tool for the semiautomated measurement of two-dimensional root architectures that makes two novel contributions. First, a top-down approach to identifying the roots themselves is described. Expectation maximization (EM; Dempster et al., 1977) clustering, to assign appearance likelihoods to pixels, is combined with an A* search running over these values to fit a model path from root apices to the seed point, thus identifying root branches. This statistical, model-led approach avoids the need to parameterize an extended series of interacting filtering operations. Though a number of model parameters are required, these are fixed for particular plant species and imaging methodologies: a set of predefined models are provided with the tool, but the RootNav interface allows experienced users to modify them or add their own. The EM stage assumes that each object of interest (root, background, etc.) can be represented as one or more pixel clusters, each having a constant gray value but affected by additive Gaussian noise. The EM algorithm is applied locally, allowing RootNav to be applied to images acquired under nonuniform illumination. The A* search assumes root branches to be smooth and continuous and prefers paths close to the root centerline.

The second contribution of this paper is an intuitive method for allowing users to guide the model-fitting process. Automatic tip detection aids the user in marking the end points of root branches, which are extracted by the A* search. Then, drawing on skills used for defining routes in familiar motorists' satellite navigation software, the user is given the ability to change paths defined from root tips to the origin point by dragging any point(s) lying on the path to a new location. The optimal path is then automatically recalculated, forcing this new location to be visited. This allows the user to guide RootNav through, for example, clumped roots, should data on these be required. RootNav's combination of user guidance of an automatic image analysis process (Lobet et al., 2011) provides fast, accurate segmentations of RSAs with a high level of objectivity.

RSAs captured in RootNav are described by a rich, nested data structure stored in a database, which can be exported in a simple, text-based format. RootNav is demonstrated to work on a variety of species (evaluated on wheat [*Triticum aestivum*] and demonstrated on rice [*Oryza sativa*], *Brassica napus*, and Arabidopsis [*Arabidopsis thaliana*]) and on images captured using a range of acquisition techniques and growth methods (digital imaging, flatbed scanning of roots grown in

soil, germination paper, and agar gel plates). Though the evaluation reported here focuses on four RSA traits, RootNav provides the structural information needed to support extraction of a wide variety of biologically relevant measures. A separate database viewer tool is provided to recover a rich set of architectural traits from RootNav's core representation.

RESULTS

Preprocessing Using EM

Rather than adopt the more common thresholding approach to foreground detection, RootNav utilizes the EM clustering algorithm (Dempster et al., 1977) to calculate the likelihoods with which a given pixel depicts a plant root, background material, or other expected objects. Likelihood values estimated at each pixel provide the raw data input to an optimal path algorithm that fits a model of an individual root.

Intensity thresholding relies on setting a value above which all pixels are immediately considered foreground (in this case, root material). While it can be effective, it is highly sensitive to noise and lighting, and often the optimum threshold value varies across the image. Adaptive methods allow different thresholds to be computed from and applied to local image regions, but the key problem remains. Threshold-based methods (be they local or global) make irrevocable decisions as to which pixels represent root material too early, on the basis of either potentially unreliable (in the case of global methods) or comparatively little (in the case of local adaptive methods) information. As a result, they are more likely to make erroneous choices than a top-down approach, which estimates the probability with which each pixel represents root material, but postpones its decision making until it has considered all the pixels that might link a specific root tip to the top of the RSA.

EM is itself a top-down method. Pixels are clustered into groups based on their intensities by fitting a Gaussian mixture model (GMM; Stauffer and Grimson, 1999) to a histogram of the data (Fig. 1). GMMs are comprised of k Gaussian components, each representing a cluster of pixels and associated with a weighting term representing its contribution to the model. Output from EM is a set of k values for each pixel, estimating the likelihood of that pixel's intensity being generated by each of the k GMM components in turn. Together, the GMM components form a measure of the likelihood that any given intensity level will occur. Figure 1C shows the GMM representation of a section of an image of a wheat root.

The assumption underlying GMMsis that each object of interest in the image can be described by one or more (Gaussian) components. However, depending on the capture technique illumination may vary significantly and systematically over the image area. This can locally skew the intensity histogram, violating this key assumption. In this case, an effective solution is to split

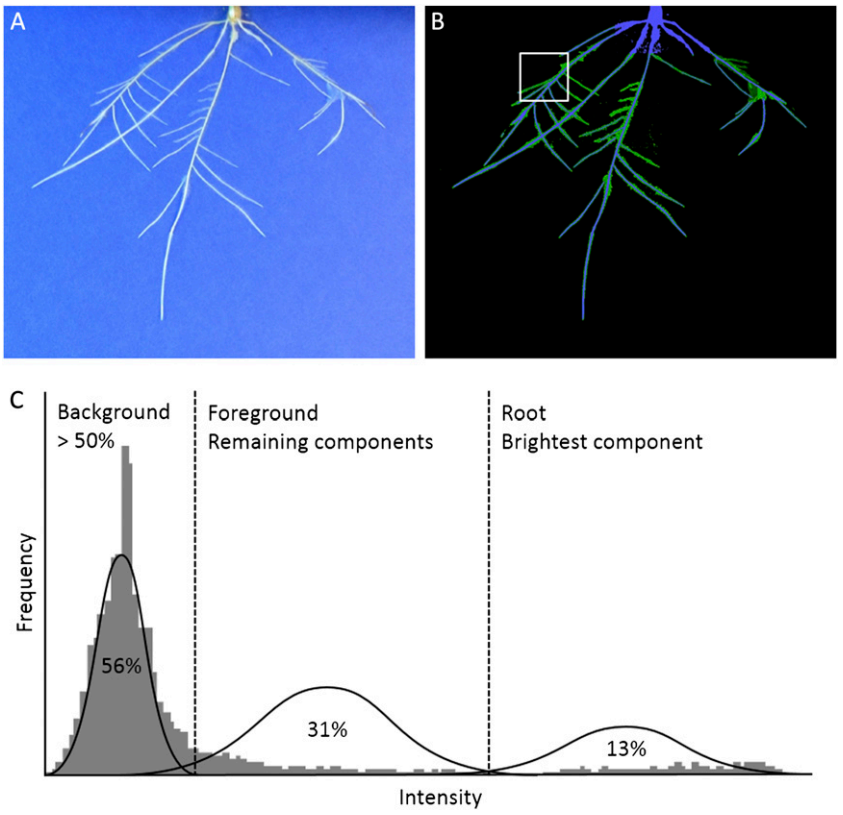

Figure 1. A, An image of a wheat root growing on germination paper. B, A classified image based on the output likelihoods of EM. Each pixel is color labeled with the component from which its intensity is most likely to have arisen, illustrating the discriminative power of the GMM. C, A histogram of the image patch marked in B. Three Gaussian components are fitted using EM, then classified into background (in this case, one class alone represents $>-50\%$ [$T = 0.5$] and so is assigned to background) and foreground. Finally, the brightest foreground component is chosen as the root component. Note the vertical scale of fitted components in C has been exaggerated for clarity.

the image into smaller subregions and apply EM independently to each. The assumption underlying this is that background illumination only varies slowly and will be approximately constant over each region, so the local GMMs will be unaffected by global illumination effects. This approach is adopted in RootNav; the original image is divided into smaller rectangular subregions. Note that this is a conservative strategy; results will be improved in the presence of global changes but unaffected if illumination is constant. While local computation of GMMs is reminiscent of local thresholding, it should be stressed that no irrevocable decisions are made on the basis of these models alone. They merely provide likelihood data to the root detection (optimal path) process.

Before a root model can be fitted to the data supplied by EM, it is necessary to determine which cluster (s) correspond to root material, i.e. which likelihood value class estimates the likelihood of a root being present, versus the likelihood of the background, etc. RootNav assumes the image will contain a dark background and a light foreground. If this is not the case, for example, in infrared images of Arabidopsis growing on agar plates, the image intensities can simply be inverted. By assuming the background will be darker, any background components in the GMM will have lower mean values; one or more components with low means can be taken as describing the background distribution.

Components are assigned to the background distribution using a technique based on the approach of Stauffer and Grimson (1999). In this work, the authors assume that a given percentage of the image will depict background and can thus make a reasonable assumption as to the number of Gaussian components required to generate it. The first B GMM components are selected as background:

$$B = argmin_b\left(\sum_{k=1}^{b} \omega_k > T\right)$$

where T is a measure of the minimum proportion of the image region that should be accounted for by the background. In Stauffer and Grimson (1999), these distributions were ordered based on their weights. In RootNav, background distributions are ordered by their mean intensities. Essentially, Gaussian components are assigned to the background representation in turn, starting with the class with the lowest mean intensity. This process continues until the sum of the weights of these background distributions is greater than T. When this occurs, any remaining distributions are considered foreground (plant). Finding a suitable value for T is a simple task; for a given region size, the approximate percentage of foreground can be estimated given the size and density of the roots in the image. Because most root systems will occupy much less than one-half of the image pixels, a T value of 0.5 works well as a conservative estimate, resulting in classes representing at least one-half of the pixels being assigned to the background cluster (see Fig. 1C). If there is only background in the region of interest (e.g. it is located in an isolated area of filter paper or gel plate, for example), RootNav will still try to apply k classes, when actually just one is required. In the case of a uniform background, this will typically result in k highly overlapping, though differently weighted, Gaussian distributions, which can be filtered out using heuristics. Typically, even without filtering, this miscategorizing of clusters does not affect the optimal path algorithm, as it most notably occurs on regions of the image that do not contain roots and so will not be traversed anyway. In rare situations where less than one-half the pixels are background and root material is present, which typically only happens near the point of root origin (i.e. the seed), the class assignment may fail; in such cases, the user is able to easily reassign classes using simple mouse clicks (see Supplemental Fig. S1).

The number of Gaussian components remaining to classify as foreground is dependent, not only on how many are assigned to the background distribution, but also on how many were used in total. For example, for three components, one of which is background, two remain as foreground and either or both might correspond to a root class. Thresholding would assign any pixels from these classes to a single foreground class as the basis for the extraction of root architecture. In RootNav, knowledge of the image type being examined can be used to further separate the foreground components. In Figure 1, a section of a wheat root has been segmented into a single background and two foreground classes. These foreground classes can be further interpreted as root and root hair, allowing further quantitative analysis. It should be noted, however, that no attempt to automatically assign meaning to foreground classes is made at present.

The choice of how many components to use in EM depends on the nature of the images being analyzed. For example, in infrared images of small Arabidopsis seedlings (Supplemental Fig. S2, the roots themselves are small and contain little detail. More than two classes, a single background and a single foreground, would yield no additional information. For wheat images at a higher resolution, it is possible to distinguish the intensity of the root from that of root hairs. In this case, an additional third component is identifiable (Fig. 1B). Given scans of rice roots (Supplemental Fig. S2), three components is a reasonable choice; however, in this case, the darker of the two foreground classes corresponds to lateral roots, which are smaller and darker in these images than in the wheat images. It is possible to configure RootNav to handle new types of images and plant varieties by adapting the number of components and other properties. RootNav also provides a configuration file with appropriate default values for image types that have been tested during development. It is anticipated that for most users, these prestored models will be sufficient.

Extracting Roots as Optimal Paths

Component probabilities now correspond to the likelihood that a given pixel represents background or foreground and provide classification information on the nature of that pixel. RootNav uses this information to produce a cost map that will be used as the basis for an A* search algorithm (Hart et al., 1968). Pixels with a high likelihood of being generated by a background component are assigned a high cost. Pixels with a high likelihood of being part of the foreground are assigned lower costs (Fig. 2B). Though high cost pixels are typically darker than those assigned low cost, cost values are not a simple function of intensity, but reflect model likelihoods.

To extract root architecture, RootNav computes a series of low-cost paths across the image, based on the per-pixel costs calculated in the previous step. This process begins with the identification of the start and end points for each root (Fig. 3A). This is a semiautomatic process in which the user selects the sources of each root system (i.e. the seed) and the primary and lateral tips of each root and is analogous to defining start and end destination in a route-planning system. This process is made faster by the inclusion of automatic tip detection that will 'snap' the user's clicks to nearby tips if possible. Details of the tip detection method and user interface are given below.

Once all necessary plant sources and root tips have been identified, the lowest-cost path between each source and each tip is produced and used to reconstruct the root architecture. RootNav uses the A* search algorithm to find a path of low cost between two arbitrary points. A* operates on a graph structure, with weighted edges between nodes. In RootNav, a graph vertex is created at each pixel, with edges between pixels defined in eight directions. Weights are assigned to these edges based on the costs calculated in the previous EM classification step (Fig. 2C). For efficiency, on larger images, it is helpful to use a graph at a lower resolution than that of the image. Creating nodes every nth pixel reduces the memory requirements of the algorithm by a factor of up to n^2. Weights are calculated as the product of the EM-derived weight

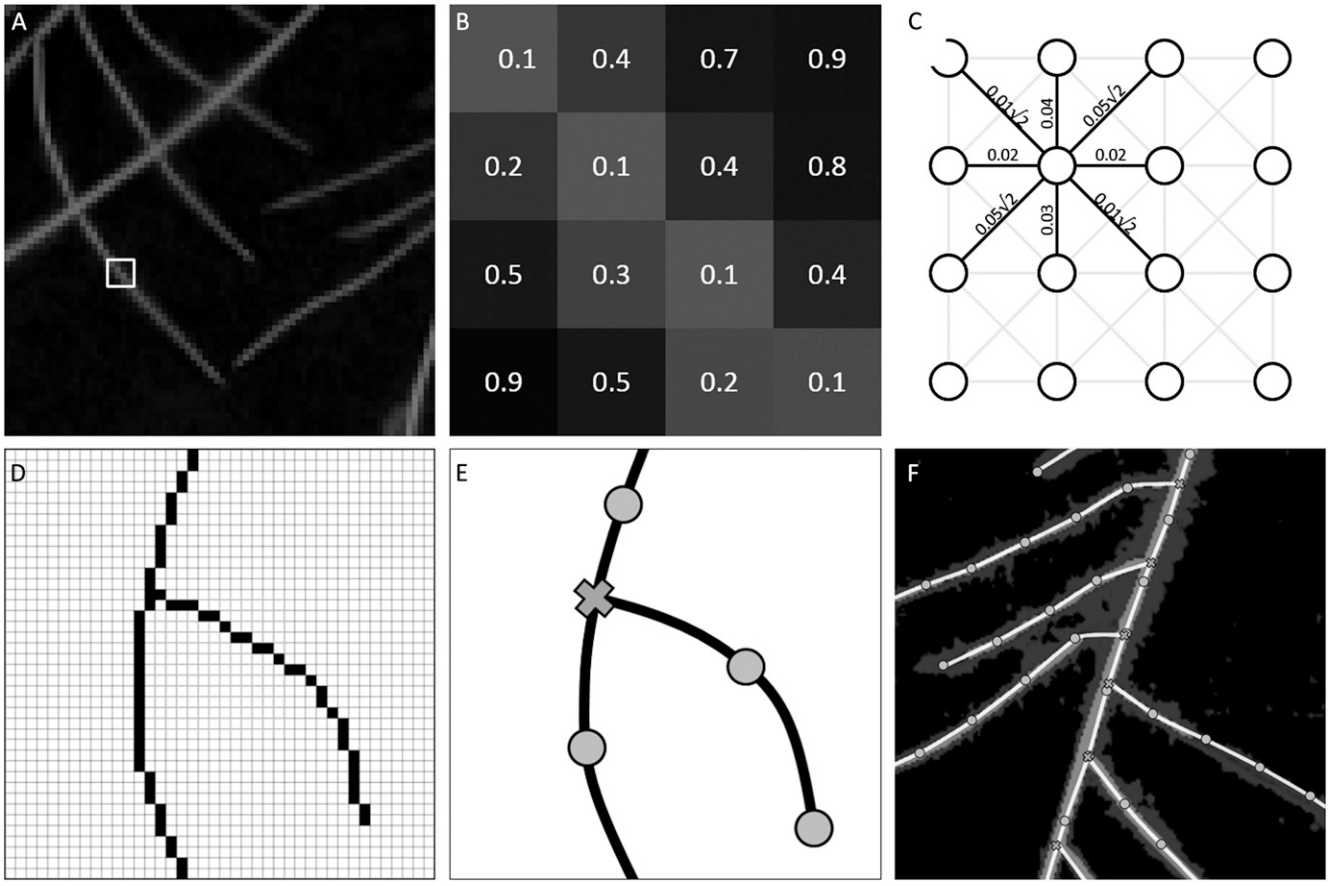

Figure 2. Example of how weights are calculated. A, Location of close-up section. B, Close-up example of pixel weights derived from the EM clustering. C, A sample of the resulting edge weights. Note that diagonal edges are weighted by $\sqrt{2}$ (Euclidean distance). D, Pixel level path along roots. E, The resulting fitted spline. Note that laterals are connected at junction points (marked with an X) and become a child (in graph terminology) of the root they are connected to. The joining of laterals to parent roots is also illustrated in F.

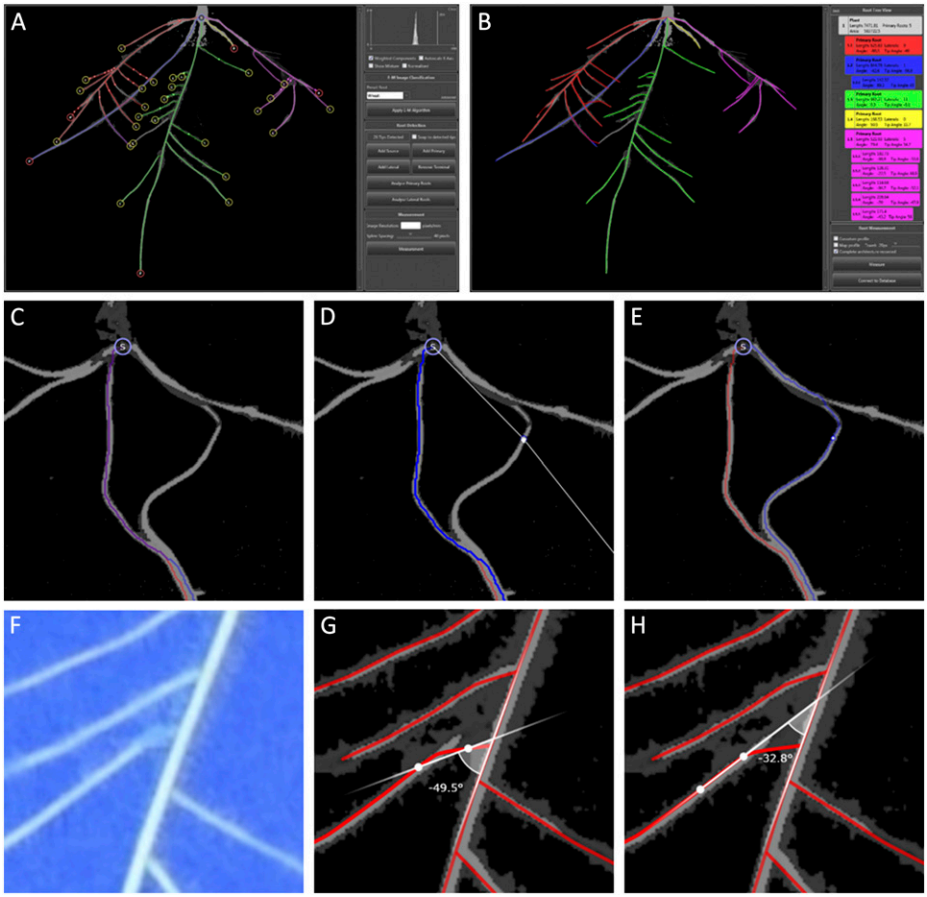

Figure 3. A, User input phase screenshot. Tips and sources represent as circles. B, Measurement phase screenshot. C, Shortest path error: both roots take the same shortest path. D, The user drags one path over to another location on the image. E, The root trace is rerouted through this position, correcting the architecture with one mouse movement. F, Example of a poor-quality image. G, A highlighted root showing angles. An angle that is not representative of the true angle of emergence due to poor image quality in that region causing the shortest path to be truncated. H, A corrected angle after the control points have been moved by the user.

of each pixel along a given edge, where diagonal edges are also multiplied by $\sqrt{2}$ to account for their additional length.

The precise weighting function used is:

$$w((i_1,j_1),(i_2,j_2)) = \begin{cases} \prod_{j=\min(j_1,j_2)}^{j=\max(j_1,j_2)} w_{i,j}, & i_1 = i_2 \\ \prod_{i=\max(i_1,i_2)}^{i=\max(i_1,i_2)} w_{i,j}, & j_1 = j_2 \\ \prod_{k=0}^{|i_1-i_2|} w_{\min(i_1 i_2)+k,\min(j_1,j_2)+k}\sqrt{2}, & otherwise \end{cases}$$

A* search is an extension of Dijkstra's algorithm (Dijkstra, 1959), which supports the addition of heuristic measures to improve efficiency. Dijkstra's algorithm is guaranteed to find the path of lowest cost between two points in weighted graph. As such, it is well suited to use as an image traversal tool and has been employed in previous work, notably NeuronJ (Meijering et al., 2004). A* search has been used effectively within the Simple Neurite Tracer (Longair et al., 2011). Both these tools are, however, unsuitable for the efficient extraction of complex RSAs. Neither allows for nested structures, such as those that exist between lateral and primary roots. When roots cross, both NeuronJ and the Simple Neurite Tracer require each path to be added in small consecutive sections to avoid an incorrect global path. This is a common occurrence in root architectures of a moderate age. RootNav avoids this problem by first computing a globally efficient path but then allowing the user to very quickly make large deviations from this path by adding control points, through which subsequent paths are forced to travel.

Dijkstra's algorithm is often precise where the globally shortest path between two points must be calculated but suffers a number of notable drawbacks that have resulted in our use of the extended A* search. Dijkstra's method is extremely efficient when traversing roots, as they have low weight and are traversed quickly while the algorithm ignores many background areas. However, when poor image quality produces breaks in the low-weight paths along a root, the search will proceed through background pixels, at which point the lack of low-weighted pixels to guide the search makes it much less efficient. A* allows the addition of a heuristic based on the distance remaining to the root source to the calculated weight of the current path; where no appropriate low-weight steps can be taken, the path will travel directly toward the end of

the root. By applying a suitably low weight to this heuristic relative to the current path cost, the algorithm is prevented from taking unnecessary shortcuts around curved root sections.

It is also possible that in thicker root sections, the shortest path may travel along a root edge, rather than the intuitively more appealing centerline: in curved sections of root, the inside edge will be markedly shorter than the outer one. To overcome this, an additional heuristic is added to A*, considering additional weights based on the distance to the center of a root mass, i.e. a distance map. This novel combination of path cost, combined with a "distance-to-go" heuristic, and a distance map heuristic allow RootNav's specific A* to extract accurate paths between root tips and sources quickly. These can then be manipulated by the user, as described below. The advantage of a shortest path between two key points approach is that the method will always find a path, breaks in the root appearance caused by clutter, shadows, growth into the medium, etc. will be bridged, despite the higher cost. This presents a marked advantage when compared with bottom-up approaches.

The root descriptions produced by the optimal path algorithm are held as lists of (x, y) pixel coordinates representing each step along the path. Construction of a smooth representation of this root architecture is preferable for visualization, eases calculation of root traits, and allows the user to manipulate whole roots rather than (smaller) image features. The pixel-based representation of each root component is therefore replaced by a spline curve, initialized using the underlying pixel data (Fig. 2, D and E). A series of spline control points are sampled, with equal spacing along each root (Fig. 2F). A cubic Hermite spline is fitted through this data, with an empirically chosen, constant-tension parameter that controls how flexible this line will be. Cubic splines are guaranteed to pass through the control points, which is essential, as these points will be sampled along the original root. The spacing of the control points can also be reduced via the user interface if a user finds that roots are not represented with the required accuracy.

User Interaction and Root Tip Detection

RootNav is a semiautomatic tool that requires the user to indicate the start and end of root branches and specify whether each is a primary or lateral root but provides support in this via an automatic tip detection algorithm. Tips are detected using a combination of a Harris corner detector (Harris and Stephens, 1988) and additional postprocessing. The Harris detector is capable of distinguishing edge features, such as areas of the image along a root meristem, from corner features, such as those seen at a root tip. The detector defines a corner as a region of the image where the intensity changes in all directions.

While the Harris detector will find many root tips (Supplemental Fig. S3), false positives may be generated in areas of noise or along noisy edges of roots. To remove these false detections, regions around each potential tip are examined and the number of roots entering or leaving that area is counted. It is assumed where a corner has been detected, and only one root enters this region, that a true tip has been found. In cases where two or more roots enter or leave the region, the feature detector is likely to have generated a false positive, and this position can be rejected (Supplemental Fig. S3). Such an exhaustive search for roots is computationally inefficient unless first targeted with a feature detector. It will be noted that root counting requires root material to be distinguished from background. This is achieved via a standard maximum-likelihood method operating over the EM likelihood data. Though EM data can be used to identify root material directly, the method is only used in tip detection, where high accuracy is not required.

When end points have been identified, the optimal path algorithm is applied and favors solutions lying on the centerline of a single root. Where roots lie close to or touch one another, however, it is possible that the optimal path may travel along more than one root. This is similar to the operation of a satellite navigation system, where a route may be found along a nearby fast, low-cost road, despite the user wishing to avoid this road. Much like a navigation system, RootNav allows the user to specify key locations along a root through which a path must travel. The interface allows the user to drag a path to an alternate area of the image. This will force RootNav to recompute the optimal path (typically <1 s, Core i7 processor), using the same model parameters but requiring it to pass through the defined location (Fig. 3, C–E). Any number of additional key points can be added at the discretion of the user. It should be noted, however, that in the majority of cases where correction is necessary, defining one or two key points will be sufficient.

Once spline curves have been fitted to the results of the shortest path walks, a nested root structure exists from which measurements can be taken (Fig. 3B). RootNav takes some measurements automatically, and in the interests of clarity and transparency, many of these measurements can be shown graphically on the interface. In most cases, this simply affords the user some confidence in the measurements, but some measurements can also be adjusted if necessary. For example, lateral root emergence angles can be adjusted if the shortest path has traveled through an area of image noise, causing a lateral-seminal junction to be mistraced; Figure 3, G and H, demonstrates this principle. In Figure 3G, poor image quality has meant that the shortest path approach has skipped a small amount of the lateral root, jumping directly to the nearest primary root instead of taking the true path. The user may adjust the calculated angle by dragging the white control points (Lobet et al., 2011), thereby changing over which part of the lateral root the angle is calculated; the interface shows this change and the adjusted angle measurement.

Quantifying Root Architecture

Following the initialization of all spline curves, RootNav will hold the complete architecture for any number of plants, primary and lateral roots. The roots are stored in a nested structure, with roots held as components of a plant and each lateral root having a parent root (Fig. 2, E and F). This information supports the computation of numerous biologically relevant measurements, including the emergence angles of laterals, tip angles, lengths, and convex hulls.

Once architectures are traced, results are stored in a local MySQL database. From these raw results, a number of architectural traits can be computed. For a complete list of measured outputs, please see Table I. Some traits can be calculated directly within RootNav (such as root angles; Fig. 3, G and H). This data can be exported into common spreadsheet formats (for use in programs such as Microsoft Excel). However, many additional traits can be extracted from RootNav's description using the included Viewer software. For experiments involving a large number of plants, measurement can be deferred until all architectures are recovered, then all plants can be quantified simultaneously. Using the root architectures stored in the MySQL database, the Viewer tool is capable of measuring a large number of traits on any number of plants, exporting these directly into data tables or CSV files. The complete list of measured outputs added in the Viewer utility is also included in Table II. The viewer can be easily extended using a simple plug-in system, allowing further traits to be added as necessary. Detailed documentation of this tool and instructions on writing plug-ins is included in the supplemental materials.

Evaluating RootNav

To demonstrate the effectiveness of the approach and methods it employs, RootNav was applied to images of 172 root systems of a variety of winter wheat accessions (see Fig. 4B for exemplar images). Growth and imaging conditions are listed in "Materials and Methods." During this experiment, user experience showed that approximately one-half of the images did not need any user interaction beyond specifying source points and tips. For the images that did need refinement, this was typically achieved in less than a minute using the drag-and-drop features of the interface. Results of the RootNav analysis are presented in Figure 4. Only four subsets of the possible measures are presented for clarity and to demonstrate the type of analysis that is possible.

The software has also been used successfully on other species (Arabidopsis, *B. napus*, and rice; for example results, see Supplemental Fig. S2).

With a new method such as RootNav, it is important to verify the results against a manual ground truth. To this end, in five images from the data set, the root architecture was manually quantified using the image analysis software Fiji (Schindelin, 2008). The numbers of seminal and lateral roots were counted, and the total length of the seminals and laterals measured. Results are presented in Table III and discussed in the next section.

DISCUSSION

When examining the RootNav output shown in Figure 4, it is first worth noting from the error bars the high precision of the results across multiple images. With respect to accuracy, the total length of the system as measured by RootNav is shown to be on average approximately 2% shorter than the manual measures. Of course, which measurement here is "correct" (the manual measure or RootNav) is a matter for debate, but this result shows them to be in close agreement.

The largest error comes from the measurement of laterals and originates from the RootNav user not tracing the very smallest laterals, hence the relatively large discrepancies on lateral count. Total root lengths can, however, still be considered accurate (there is still only a –2.8% error on the highest lateral count discrepancy). This kind of error will not affect other geometric measures such as convex hull, as the laterals that are missed in RootNav are very small. Note that deciding to trace these small laterals is very much a subjective judgment (they may only be 5 or less pixels

Table I. *Example output measures possible with RootNav*

Trait	Unit	Description
Total length	Pixels or mm	The total length of root material for a given plant or given root in a plant
Tip angle	Degrees from vertical	The angle of a given root tip calculated relative to the *y* axis of the image
Emergence angle	Degrees	The angle of emergence for a lateral root, calculated relative to the primary root to which it is attached
Start distance	Pixels or mm	The distance in pixels or mm along a primary root that a given lateral root has emerged
Convex hull area	$Pixels^2$ or mm^2	The area of the convex hull that encompasses all rot material belonging to a given plant
Child count		The number of child roots a given root has
Classification map profile	Frequency over length	A two-dimensional data set showing the frequency of occurrences of a given classification in EM, over changes in depth; this can be used to measure the amount of root hair along a root or the amount of lateral roots along a rice seminal root

Table II. *Example output measures possible with the RootNav viewer tool, operating on a database of previously reconstructed root architectures*

Trait	Unit	Description
Total length	Pixels or mm	The total length of all root material; can be calculated separately for each plant and each root within a plant
Average length	Pixels or mm	The average length of all roots in a given plant; can calculate separate values for all roots or just primary or lateral roots
Primary root count		The number of primary roots in a given plant
Lateral root count		The number of lateral roots in a given plant; can also calculate the number of lateral roots for a given primary root
Convex hull area	$Pixels^2$ or mm^2	The area of the convex hull that encompasses all rot material belonging to a given plant
Tortuosity		The true length of a given root divided by the Euclidean distance from the source to the tip of that root; an average of this value for all roots in a given plant can also be calculated
Maximum width	Pixels or mm	The maximum width of the root system for all depths; only points at equal depths are considered
Maximum depth	Pixels or mm	The maximum depth achieved by the root system
Network aspect ratio		The maximum width of the root system divided by the maximum depth; providing a measure of the aspect ratio of the root system
Width profile	Pixels or mm over length	A two-dimensional data set showing the measurement of the width of a plant root architecture over changes in depth
Curvature profile	Degrees over length	A two-dimensional data set showing the angle of curvature of the root over changes in depth

long and very faint). This is true also of the missed seminal in Table III (image 1); this was short and overlapping another seminal, making them hard to identify. The error, then, is likely to be one of intersubject variability; using RootNav (versus entirely manual methods) will provide consistency to the results obtained.

The winter wheat results demonstrate the ability of RootNav to highlight differences in the four example traits presented (Fig. 4) in a relatively high throughput and, compared with manual alternatives, an objective manner. Clear differences can be seen for example between the modern Glasgow variety and the wild relatives (A1, A2, and A3).

In addition to numerical results, qualitative usability inferences can be drawn. First, in the example presented, only around one-half of the images needed manual intervention, and where this was required, it only took a minute or less to work the data up in the majority of cases. The time required to generate all the results from manual measurement is much greater than the time required when using RootNav. Additionally, the interface and design approaches used in RootNav should help to reduce user fatigue when compared with performing the equivalent number of measures manually in nonspecialist software. While appreciating that semiautomatic systems cannot be truly objective, we would argue that the automatic component greatly increases objectivity, while allowing the user to guide processing in situations that cannot be handled by fully automatic methods.

It will be noted that user interventions are often made in response to crossing roots. Root crossovers are, however, less of a problem for RootNav than for bottom-up methods or techniques that search sequentially from one end of the root, looking for the other, and so risk following the wrong path and becoming irretrievably lost. RootNav's strategy of locating the beginning and end of the branch and then finding the optimal path between them means that in many cases, it is unaffected by crossing roots. While most manual interventions deal with crossovers, most crossovers do not require manual intervention.

As with all approaches that incorporate an automated component, there are some necessary trade offs that should be acknowledged. When extracting architectural traits, it is desirable that root segmentation identifies the centerlines. In RootNav, the resultant path is optimal in some sense and will tend toward the centerline due to the use of the distance map in the A* algorithm. It is not a certainty, however, that the reported path will coincide entirely with the centerline in all cases. In practice, we have not found this potential disparity to be significant. The root regions in architecture images are only a small number of pixels in width and much longer than they are wide; there is little room for error. The potential to miss curvature information in a root expressing a subtle waviness is again limited by the relatively small width of the root compared with any curvature effect.

RootNav has been applied to a range of images spanning the commonly used growth and imaging environments, including scanned root-washed samples not grown under laboratory conditions. RootNav's semiautomatic approach means that if the entire architecture must be described, the time required to analyze a root increases with root complexity and plant age. This is mitigated slightly by improved performance of the tip detector as roots become thicker. It will also be noted that in the examples given here, the whole root system is in view. This is not necessary; RootNav can be used to extract partial descriptions from partial views, and the combination of descriptions obtained from partial views of very large roots (Lobet and Draye, 2013) is aided by the use of a shortest path algorithm. The optimal path between two points on different but adjacent images can be obtained by concatenating paths from each point to a shared location on the boundary between views.

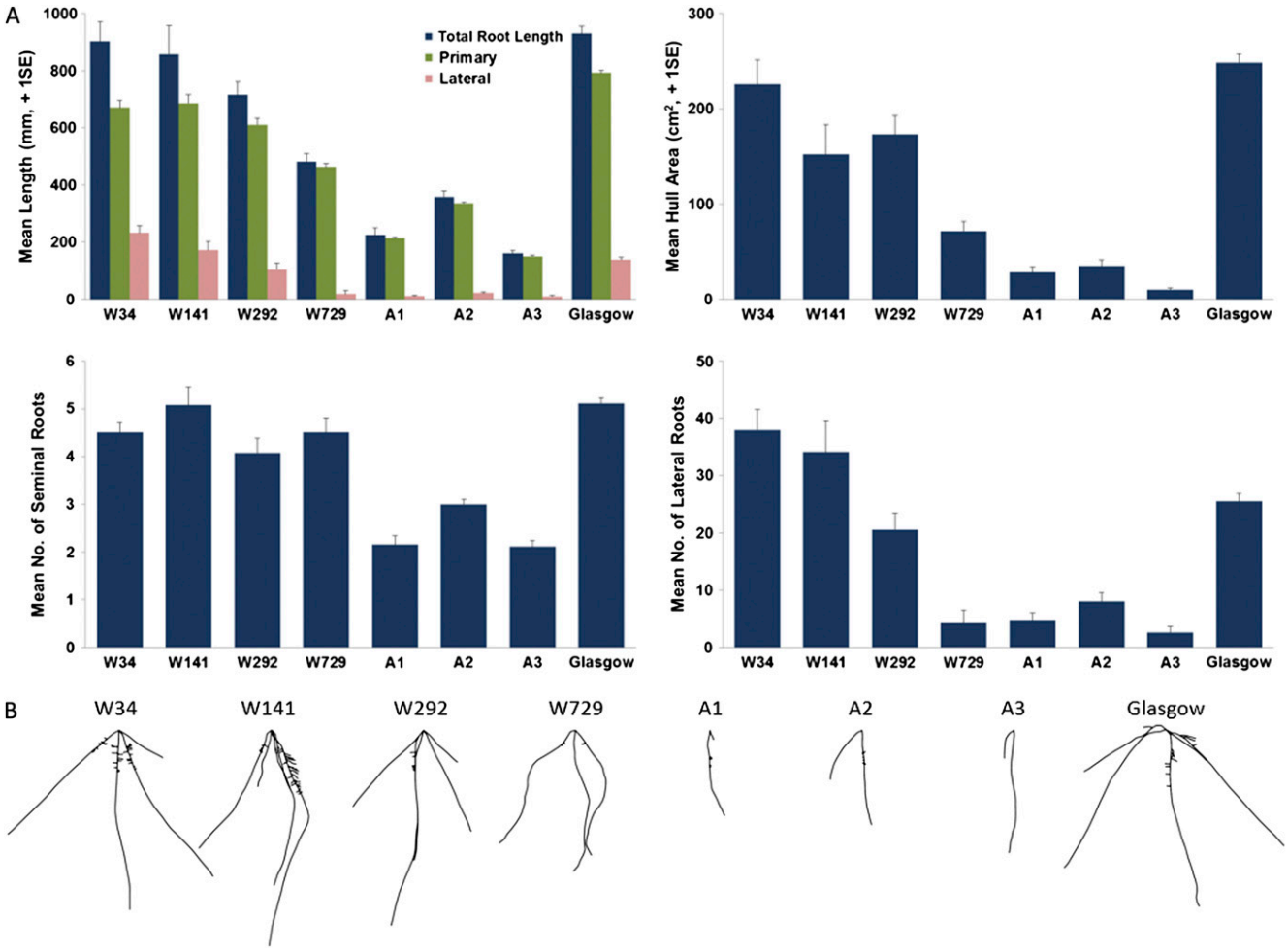

Figure 4. A, Selected RootNAV outputs for winter wheat seedlings. W34, W141, W292, and W729 refer to wheat accessions from the A.E. Watkins landrace collection sourced from India, China, Cyprus, and Iran, respectively. A1 to A3 refer to wild relatives of modern wheat (A1: *Aegilops speltoides*, A2: *Triticum uratu*, and A3: *Thinopyrum intermedium*). Glasgow refers to commercial cultivar of winter wheat. Results are means of data from eight to 18 images for the A.E. Watkins and wild relative seedlings and 80 images for cv Glasgow. Error bars represent 1 SE. B, Exemplar thumbnail images from each winter wheat accession. [See online article for color version of this figure.]

RootNav, like other systems, is controlled via a set of numerical parameters: choosing these can be difficult for the nonspecialist user. RootNav's top-down approach, however, means that its parameters define a root model. As a result, they are more intuitive, and so easier to manipulate, than parameters that control specific image analysis processes such as thresholding and thinning. Importantly, they are also more stable over a wider range of species and imaging situations and so should require less manipulation by the user. During its development, RootNav has been applied to a variety of plant species and image types, and a set of templates has been constructed, each giving parameters suitable for a specific species/environment pair.

Table III. *Results of a comparison between five manually measured architectures and RootNav*

Image No.	No. of Seminals	No. of Laterals	Length of Seminals	Length of Laterals	Total Length
	ground truth/RootNav			*% error*	
1	6/5	12/8	–1.4	–36.8	–2.8
2	4/4	15/12	–0.6	–16.0	–2.3
3	4/4	21/21	–1.2	–12.4	–4.1
4	5/5	27/27	0.0	–3.1	–0.4
5	4/4	9/9	–0.4	–6.2	–1.0

We anticipate the user's normal interaction with RootNav's parameters will be to select one of these models and not to consider (or set) individual parameters at all. The "Advanced" tab on the interface does, however, allow more experienced users to modify the prestored models and add their own should that be necessary. The user documentation supplied explains clearly what each parameter refers to and the anticipated effect of any changes.

An increasing number of root image analysis tools are now available. RootNav can be thought of as a generalization of Roottrace. Both are top down and use shortest path methods, though in Roottrace, the graph structure within which the optimal path is sought is obtained by a root tracking stage. This eases the tip detection problem but limits Roottrace to younger seedlings. Rootrace was also designed specifically for time series images of Arabidopsis grown on agar plates.

RootNav's semiautomatic approach is perhaps closest to that adopted in SmartRoot (Lobet et al., 2011). SmartRoot's user initiates and guides the search for roots by selecting points along each root branch. The system searches away from these points, attempting to locate the rest of the branch. SmartRoot produces a layered data structure based on linear sequences of straight line segments (vectors). Tools are provided that allow the user to manipulate the (many) individual vectors describing each root. By contrast, RootNav's user manipulates whole roots, with each change leading to the optimum path being recalculated. Each tool has strengths and weaknesses. RootNav produces nested root architecture, easing computation of relative measures such as root emergence angle, but is limited to single images. SmartRoot describes each branch independently but can process time series images to characterize root growth.

Other root image analysis tools, e.g. GiaRoots (Galkovskyi et al., 2012), take a very different approach, aiming more toward fully automatic, high-throughput analysis. Rather than providing an accurate, detailed structural description of a given root architecture, from which a wide variety of traits can be computed, these tools provide the summary statistics needed in large-scale experiments. Further tools exist at the other extreme of the interaction spectrum, providing a purely manual functionality, e.g. DART (Le Bot et al., 2010). DART relies on the power of the human visual system to generate the data and the power of the computer to structure that data post capture. Some tools offer a three-dimensional capability, e.g. RootReader3D (Clark et al., 2011), which are particularly good for examining effects such as circumnutation. However, the image capture stage is more complex than that needed by the easily adoptable, widely applicable system described here.

The generally applicable, powerful tools used in RootNav have of course been used in previous work. Shortest path methods have, for example, been used to trace neurons in two- and three-dimensional image stacks (Meijering et al., 2004; Longair et al., 2011). Simple Neurite Tracer (Longair et al., 2011) uses a shortest path approach to tracing between user-selected points, requiring the user to confirm created segments and respecify points as required. An alternative shortest path approach, termed Livewire, is used in NeuronJ (Meijering et al., 2004). Path costs are precomputed across the image, allowing the user to see an interactive rendering of a trace guided by the position of their mouse. The related Livewire segmentation algorithm (Barrett and Mortensen, 1997) has been used within root analysis software looking at spatiotemproal growth patterns (Basu and Pal, 2012). These approaches use variations of the shortest path approach adopted here but with important differences. First, the application domain is different; RootNav is designed specifically to address issues encountered in root architecture imaging and provides numerical results supporting the analysis of root systems. The use of the A* algorithm and the novel use of a distance map in the cost function helps RootNav to follow the centerline of the root. Second, RootNav uses an EM-based prior evaluation stage that allows a 'root-like' cost to be combined with the distance map cost. Third, RootNav allows the user to interactively reroute paths after the initial trace is made. We have found this to be a very intuitive approach for those not familiar with automated segmentation methods, largely due to familiarity with the satellite navigation tools that are so commonplace today.

CONCLUSION

In this paper, we have presented RootNav, a novel, semiautomated approach to the otherwise laborious and subjective process of extracting root architectural information from images. The two main novel components are the combination of EM-clustering and shortest path methods to segment the roots and an intuitive user interface. From the resulting measures, it is possible to extract information on a variety of traits.

An example application that used RootNav to extract architecture from winter wheat images demonstrated the effectiveness of the software. Length results are consistent with manual measures, and RootNav has been found to be both faster to use and easier to use than manual methods that produce similar results.

The image analysis methods underlying RootNav are, we believe, key to the success of the software. There are a number of key benefits afforded by the use of a shortest path algorithm, rather than the more common skeletonization approach. When traversing a skeleton, breaks in the root structure are challenging to overcome. If an algorithm reaches the end of a root, it must decide if that location is a root tip or if it is a broken root where additional root structure can be found some number of pixels away. A shortest path algorithm will always find a path from a root source to a tip, regardless of breaks in the structure, at the small cost of requiring user input to define start and end locations. If the resulting path traverses background pixels (gaps in the root) with a high weight, this simply

increases the overall cost of the path but does not change significantly the route it follows.

The EM clustering/shortest path combination approach also offers benefits in handling noise in images. When using binary thresholding, if some regions in the image are unusually noisy, these regions can be incorrectly classified as root if they are not either removed by filtering or removed by the user during some interactive process. Many existing software packages will use mathematical morphology to erode noise from the image, along with traditional filtering algorithms prior to thresholding (e.g. EZ-Rhizo; Armengaud, 2009). Any remaining noise will be discarded by the user during a manual refinement step. Software packages such as EZ-Rhizo also place strict conditions upon the nature of the input images that can be used. Any deviation from these rules will result in unreliable output. Other automated approaches will attempt to remove as much noise as possible and then quantify the remaining noise along with the root, the assumption being that the amount of noise is insignificant compared with the root size. While we believe that offering a high-throughput solution at the expense of some reliability is useful, there are times where accuracy must be preserved. In addition, including any amount of noise or inaccurate measurements without examining closely the nature of this noise increases the uncertainty of the results.

Here, we show that the RootNav semiautomated approach, which demands a small amount of interaction from the user in exchange for high accuracy and reliability, produces detailed and encompassing statistics about root architecture. The advantages of adding user interaction have been seen before (e.g. SmartRoot; Lobet et al., 2011); here, the process is made even more intuitive for the user by adding familiar approaches from route planning applications into the interaction process. The combination of top-down image analysis algorithms, intuitive user interfaces, and flexible trait measurement makes RootNav a powerful open-source software tool for plant scientists.

MATERIALS AND METHODS

Plant Growth and Imaging

Rice (*Oryza sativa japonica* 'Dongjin') plants were grown for 28 d in a polyethylene column (55-mm diameter and 150-mm length) filled with loamy sand soil (Newport Series, Food and Agriculture Organization of the United Nations Class Brown soil) at a packing density of 1.2 cm^3 cm^{-3}. Columns were saturated from the bottom upwards with deionized water and kept in a controlled environment chamber with a 12-h light/dark cycle at 28°C during light and 20°C during dark photoperiods and a photon flux density of 250 μmol m^{-2} s^{-1}. At 28-d growth, soil was gently washed away from the intact root system with deionized water, and the root system was scanned on a flatbed scanner (Regent Instruments).

Arabidopsis (*Arabidopsis thaliana* ecotype Columbia) seeds were surface sterilized and sown on vertical 125- × 125-mm-square petri plates as detailed previously (Holman et al., 2010). Each plate contained 60 mL one-half-strength Murashige and Skoog (Murashige and Skoog, 1962) media (Sigma) solidified with 1% (w/v) agar. After 2 d at 4°C, plates were transferred to controlled environment chambers at 23°C with a 12-h light/dark cycle and a photon flux density of 150 μmol m^{-2} s^{-1}. Plants were imaged under near infrared illumination at 14 d after germination as detailed in Wells et al. (2012).

Seeds of a double-haploid mapping population of *Brassica napus* (cv Tapidor × Ningyou 7) were sterilized and sown in 240- × 240-mm-square dishes on modified Murashige and Skoog (Murashige and Skoog, 1962) media as detailed in Shi et al. (2012). Root systems were scanned 12 d after sowing using a flatbed scanner (Scanjet 3670, Hewlett-Packard).

Wheat (*Triticum aestivum*) seeds (see Fig. 4 legend for accession details) were germinated and transferred to vertically arranged germination paper (Anchor Paper Co.), moistened from the base with one-quarter-strength Hoagland's nutrient solution (Sigma). Seedlings were grown for 11 d in a controlled environment chamber with a 12-h light/dark cycle at 20°C during light and 15°C during dark photoperiods and a photon flux density of 400 μmol m^{-2} s^{-1}. Roots were imaged using a Nikon D5100 DSLR camera.

Manual Ground Truth

Manual ground truths of root length measures in a subset of the images was measured in the image analysis software Fiji (http://www.fiji.sc). The segmented line tool was used to trace the roots, and then Fiji's measure command used to measure distance (in pixels).

Software Implementation

RootNav has been written in C# using the .NET framework libraries. The tool runs under Windows XP, Vista 7 and 8. Database access is achieved using the MySQL Connector library. RootNav is an open-source software project, released under the Berkeley Software Distribution license.

Supplemental Data

The following materials are available in the online version of this article.

Supplemental Figure S1. Redistribution of background components after E-M to improve classification.

Supplemental Figure S2. Images demonstrating the expected performance of the Harris feature detector, and the improved root tip detection in RootNav.

Supplemental Figure S3. Example images and results when using RootNav on a variety of plant species and image types.

ACKNOWLEDGMENTS

We thank Susan Zappala and Martin Broadley for the provision of images.

Received May 17, 2013; accepted June 2, 2013; published June 13, 2013.

LITERATURE CITED

Armengaud P (2009) EZ-Rhizo software: the gateway to root architecture analysis. Plant Signal Behav **4:** 139–141

Barrett WA, Mortensen EN (1997) Interactive live-wire boundary extraction. Med Image Anal **1:** 331–341

Basu P, Pal A (2012) A new tool for analysis of root growth in the spatio-temporal continuum. New Phytol **195:** 264–274

Bengough AG, Gordon DC, Al-Menaie H, Ellis RP, Allan D, Keith R, Thomas WTB, Forster BP (2004) Gel observation chamber for rapid screening of root traits in cereal seedlings. Plant Soil **262:** 63–70

Clark RT, MacCurdy RB, Jung JK, Shaff JE, McCouch SR, Aneshansley DJ, Kochian LV (2011) Three-dimensional root phenotyping with a novel imaging and software platform. Plant Physiol **156:** 455–465

de Dorlodot S, Forster B, Pagès L, Price A, Tuberosa R, Draye X (2007) Root system architecture: opportunities and constraints for genetic improvement of crops. Trends Plant Sci **12:** 474–481

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc, B **39:** 1–38

Dijkstra EW (1959) A note on two problems in connexion with graphs. NUMERISCHE MATHEMATIK **1:** 269–271

French A, Ubeda-Tomás S, Holman TJ, Bennett MJ, Pridmore T (2009) High-throughput quantification of root growth using a novel image-analysis tool. Plant Physiol **150:** 1784–1795

Galkovskyi T, Mileyko Y, Bucksch A, Moore B, Symonova O, Price CA, Topp CN, Iyer-Pascuzzi AS, Zurek PR, Fang S, et al (2012) GiA Roots: software for the high throughput analysis of plant root system architecture. BMC Plant Biol **12:** 116

Gasch CK, Collier TR, Enloe SF, Prager SD (2011) A GIS-based method for the analysis of digital rhizotron images. Plant Root **5:** 69–78

Harris C, Stephens M (1988) A combined corner and edge detector. In Proc. of Fourth Alvey Vision Conference. pp 147–151

Hart P, Nilsson NJ, Raphael B (1968) A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics **4:** 100–107

Hetz W, Hochholdinger F, Schwall M, Feix G (1996) Isolation and characterization of rtcs, a maize mutant deficient in the formation of nodal roots. Plant J **10:** 845–857

Holman TJ, Wilson MH, Kenobi K, Dryden IL, Hodgman TC, Wood AT, Holdsworth MJ (2010) Statistical evaluation of transcriptomic data generated using the Affymetrix one-cycle, two-cycle and IVT-Express RNA labelling protocols with the Arabidopsis ATH1 microarray. Plant Methods **6:** 9

Le Bot J, Serra V, Fabre J, Draye X, Adamowicz S, Pagès L (2010) DART: a software to analyse root system architecture and development from captured images. Plant Soil **326:** 261–273

Lobet G, Pagès L, Draye X (2011) A novel image-analysis toolbox enabling quantitative analysis of root system architecture. Plant Physiol **157:** 29–39

Lobet G, Draye X (2013) Novel scanning procedure enabling the vectorization of entire rhizotron-grown root systems. Plant Methods **9:** 1

Longair MH, Baker DA, Armstrong JD (2011) Simple Neurite Tracer: open source software for reconstruction, visualization and analysis of neuronal processes. Bioinformatics **27:** 2453–2454

Lynch J (1995) Root architecture and plant productivity. Plant Physiol **109:** 7–13

Meijering E, Jacob M, Sarria JC, Steiner P, Hirling H, Unser M (2004) Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. Cytometry A **58:** 167–176

Murashige T, Skoog F (1962) A revised medium for rapid growth and bio assays with tobacco tissue cultures. Physiol Plant **15:** 473–497

Naeem A, French AP, Wells DM, Pridmore TP (2011) High-throughput feature counting and measurement of roots. Bioinformatics **27:** 1337–1338

Pound M, French A, Wells D, Bennett M, Pridmore T (2012) CellSeT: novel software to extract and analyze structured networks of plant cells from confocal images. Plant Cell **24:** 1353–1361

Pridmore T, French A, Pound M (2012) What lies beneath: underlying assumptions in bioimage analysis. Trends Plant Sci **17:** 688–692

Schindelin J (2008) Fiji Is Just ImageJ (batteries included). ImageJ User and Developer Conference

Shi L, Shi T, Broadley MR, White PJ, Long Y, Meng J, Xu F, Hammond JP (2012) High-throughput root phenotyping screens identify genetic loci associated with root architectural traits in *Brassica napus* under contrasting phosphate availabilities. Ann Bot (Lond)

Stauffer C, Grimson WEL (1999) Adaptive background mixture models for real-time tracking. IEEE Comput Soc **2:** 246–252

Trachsel S, Kaeppler S, Brown K, Lynch J (2011) Shovelomics: high throughput phenotyping of maize (*Zea mays* L.) root architecture in the field. Plant Soil **341:** 75–87

Wells DM, French AP, Naeem A, Ishaq O, Traini R, Hijazi HI, Bennett MJ, Pridmore TP (2012) Recovering the dynamics of root growth and development using novel image acquisition and analysis methods. Philos Trans R Soc Lond B Biol Sci **367:** 1517–1524