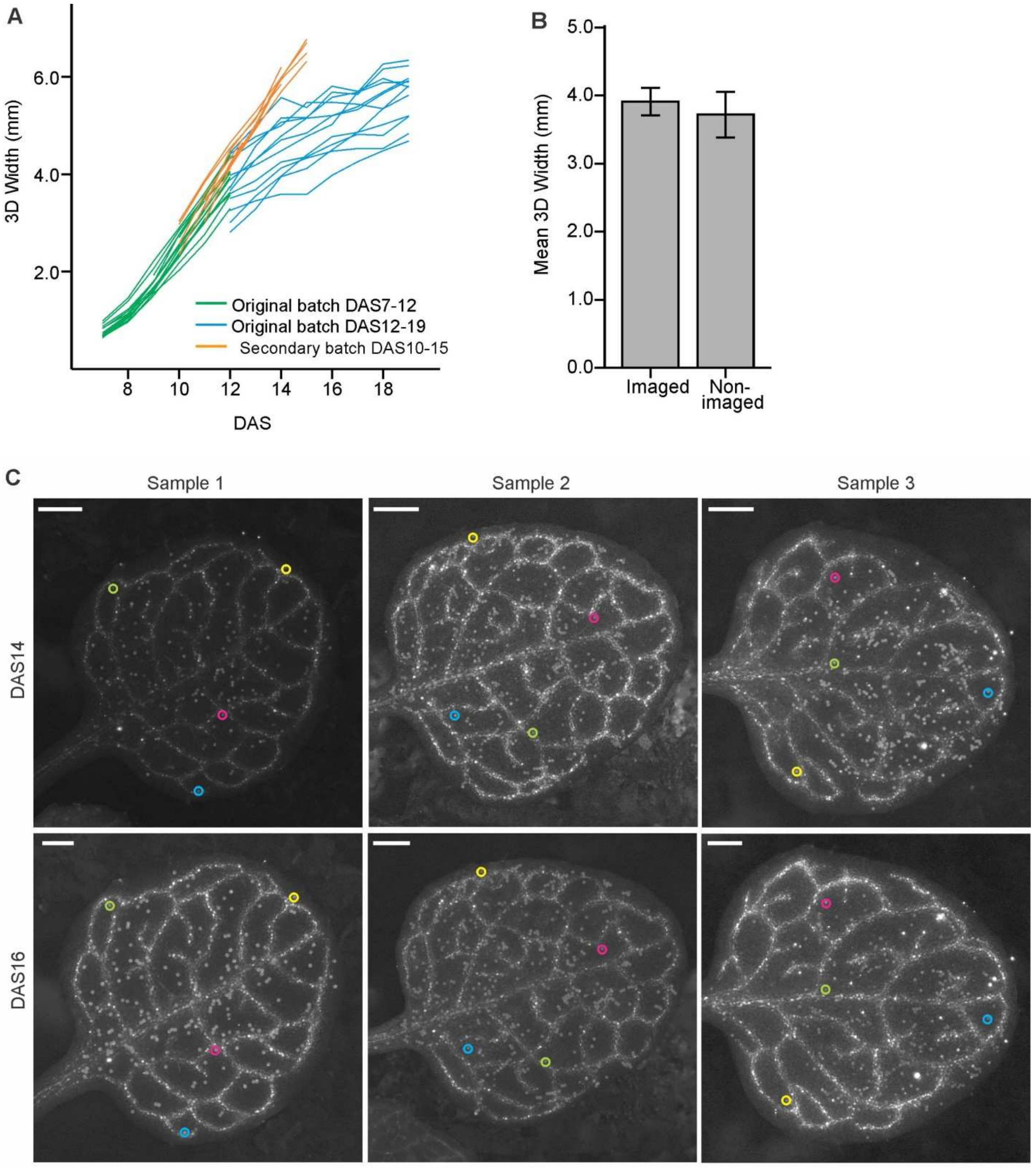

A
3D Width (mm)
6.0
4.0
2.0
8
10
12
14
16
18
DAS
Original batch DAS7-12
Original batch DAS12-19
Secondary batch DAS10-15
B
Mean 3D Width (mm)
5.0
4.0
3.0
2.0
1.0
0.0
Imaged
Non-imaged
C
Sample 1
Sample 2
Sample 3
DAS14
DAS16

Figure S1, related to Figure 1. Details on data set and validation of methods to show that imaging and bead application does not disturb plant growth and that beads do stick to the leaf surface.

(A) 3D leaf width of all samples in the data set over time. The green and blue lines represent the original batch of plants that were imaged, with a set of plants being imaged from DAS7-12 (green lines) and a set of plants being imaged from DAS12-19 (blue lines). The orange lines represent a secondary batch, grown a few weeks later in the same conditions. The data from this secondary batch was matched to the growth curves of the original data set (see Supplemental Materials and Methods for more information).

(B) To investigate the effect of daily imaging on leaf growth, we compared the 3D width of leaves that underwent bead application at DAS7 and daily imaging under brightfield and fluorescent light until DAS12 (N=14), to DAS12 leaves grown concurrently in the same conditions that had not been previously imaged (N=12). Error bars represent the 95% confidence interval. A Mann-Whitney U test reveals that there is no significant difference in the leaf width of the two groups (U=67.0, Z=-0.874, P=0.382), confirming that particle application and imaging does not disturb leaf growth.

(C) Fluorescent particles were applied to the leaves of a strain of Arabidopsis that has green fluorescent vasculature (AtSuc2prom:GFP), and leaves were imaged at two time points two days apart. Here we show three samples, and highlight the position of a few beads from one time point to the next by coloured circles. The particles remain in the same relative positions compared to the vasculature, demonstrating that they remain fixed to the leaf surface and move with it as the tissue expands. We used fluorescent red beads which still fluoresced weakly under the GFP filter, since the vasculature fluorescence was very weak compared to and overpowered by the yellow fluorescent beads. Scale bars represent 1mm.

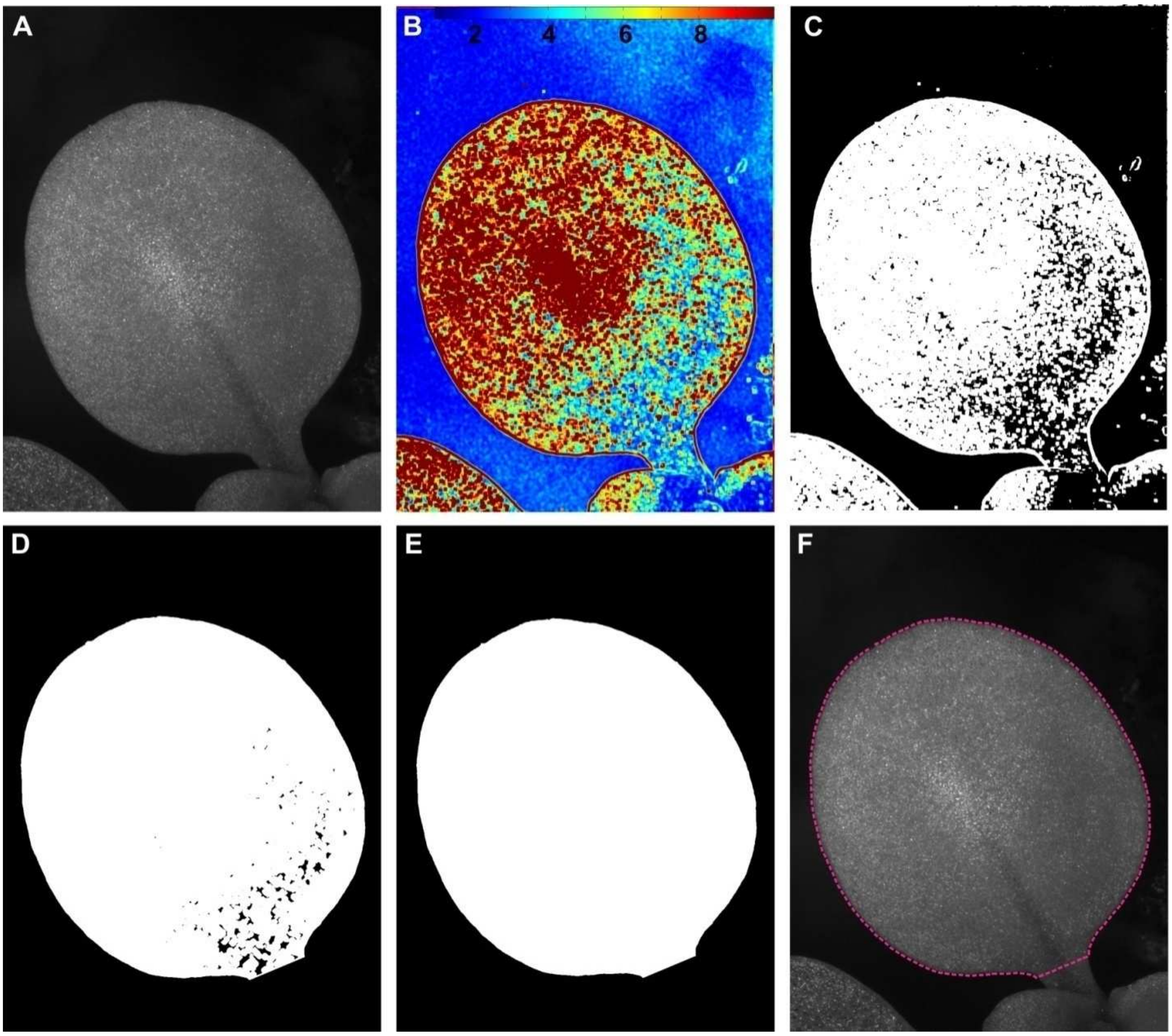


Figure S2, related to Figure 1. Details of the automated leaf-outlining software.

(A) The leaf outline is extracted from the brightfield multi-focus mopntage image. In this example we show a leaf at DAS11.

(B) We first apply the Matlab *stdfilt* function (built-in Matlab functions are cited in italics throughout) from the Image Processing Toolbox to the brightfield multi-focus image (A), which returns at each pixel the local standard deviation of the image (in a window of 9x9 pixels). Areas outside of the leaf typically have low standard deviation as they are mostly dark, and out of focus (i.e. there are no sharp changes in pixel values). Areas within the leaf have higher standard deviation due to the texture and shininess of cells on the leaf surface, and standard deviation is especially high at the boundary between the leaf and surrounding soil as the soil is much darker than the leaf surface, giving a large jump in pixel values at the leaf perimeter.

(C) A new binary image is created from areas where standard deviation (B) meets a minimum threshold (5.0), giving the outlines and some regions of the surface of any leaves in the image. The *imdilate* function is applied to smooth edges slightly and link any regions of the outline that are not connected.

(E) The program uses the left and right petiole coordinates to create a line that segments the image along the petiole; the region that does not contain the leaf tip is cropped out. A line with the left and right petiole coordinates as end points is added to the binary leaf image, to close the leaf outline along the petiole. To eliminate other leaves in the image or small objects in the soil, we apply the *bwconncomp* function, which identifies objects in an image based on pixel interconnectivity. The largest object in the image is the leaf in question, and all other objects are discarded.

(F) Now we apply the *imfill* function to fill in all enclosed areas in the image, to produce a single object corresponding to the entire leaf surface, followed by the *imerode* function to shrink the leaf object slightly since it was expanded slightly when the *imdilate* function was applied in an earlier step.

(G) We then apply the *bwmorph* function with the ‘erode’ operation to leave only the outline of the leaf object. The *bwtraceboundary* function can now be applied to extract x,y-coordinates of this image. This line is plotted on the original brightfield image for the user to verify.

On our computers this software computes and displays the predicted leaf outline in under 3 seconds.

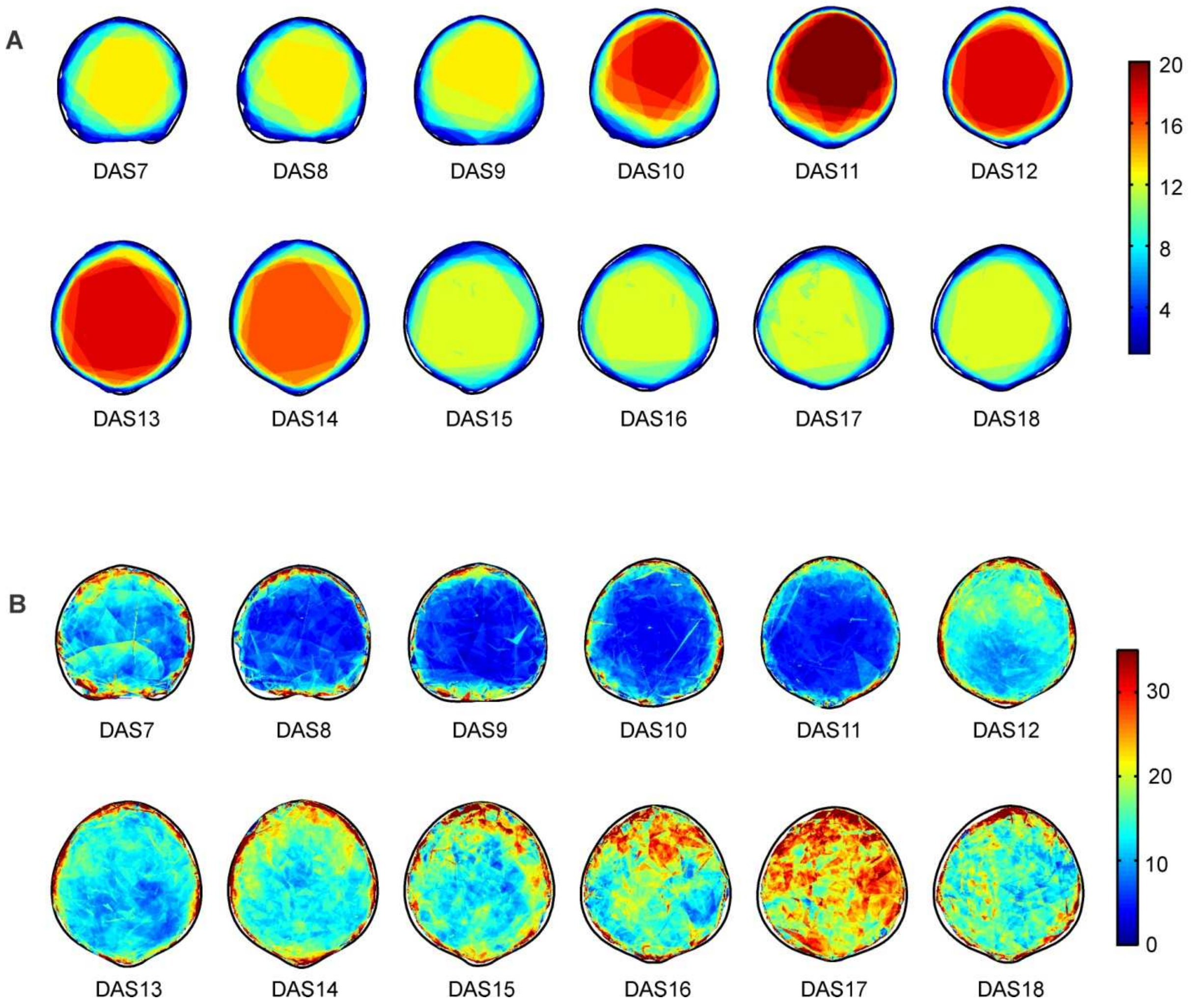


Figure S3, related to Figure 2. Spatial maps of sample numbers used in the calculation of the mean maps and the relative standard error (RSE) of the RG maps.

(A) Number of samples used to compute the mean values at each point on the mean spatial maps, coloured according to the scale shown on the right.

(B) Relative standard error (RSE) of the mean RG maps, coloured according to the scale shown on the right. RSE is expressed as a percentage, e.g. if the error at a point is 30% RSE and the mean growth at the same point is 20% RG, the RG at that point is 20±6% (30% of 20% is 6%).

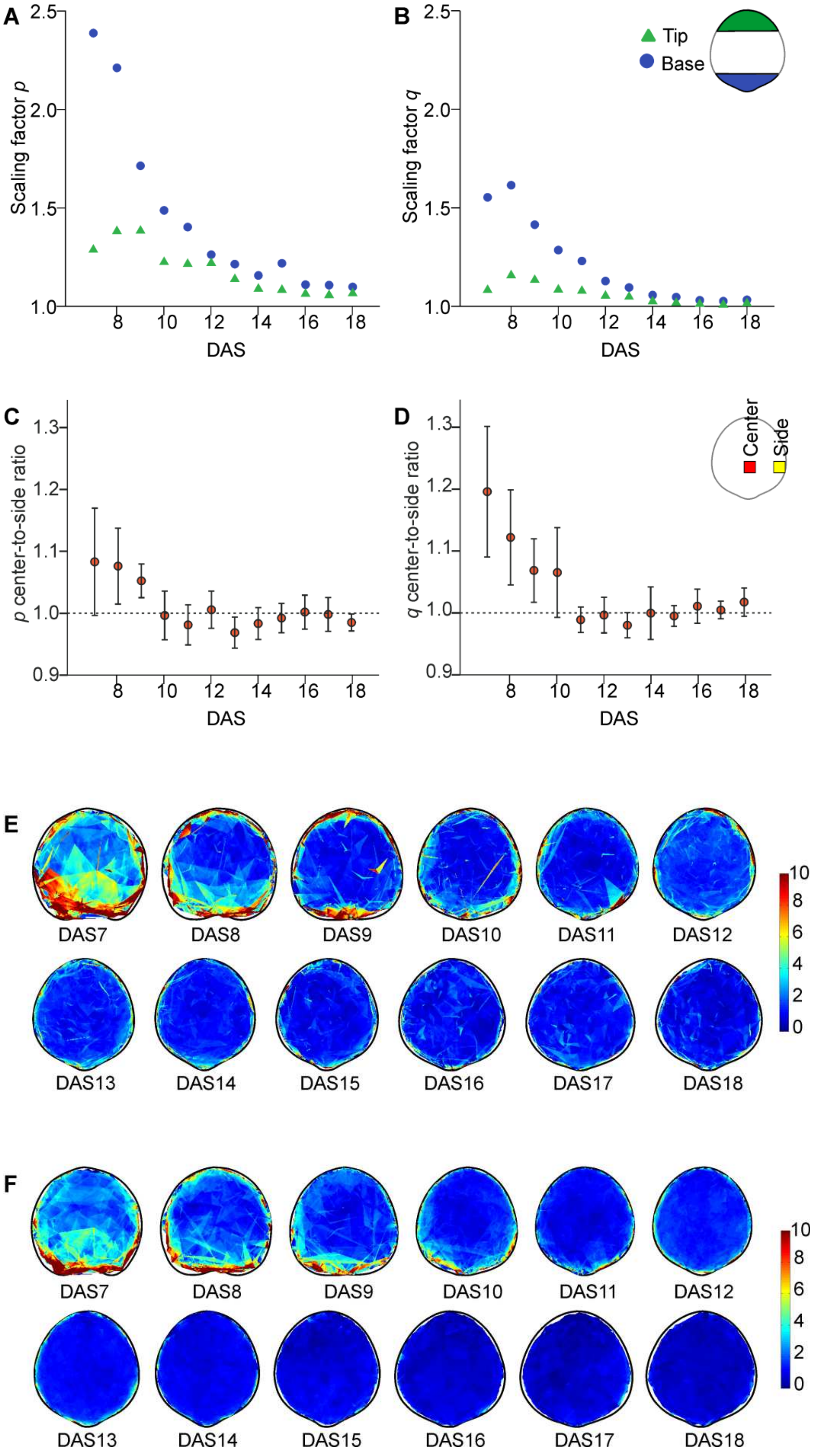
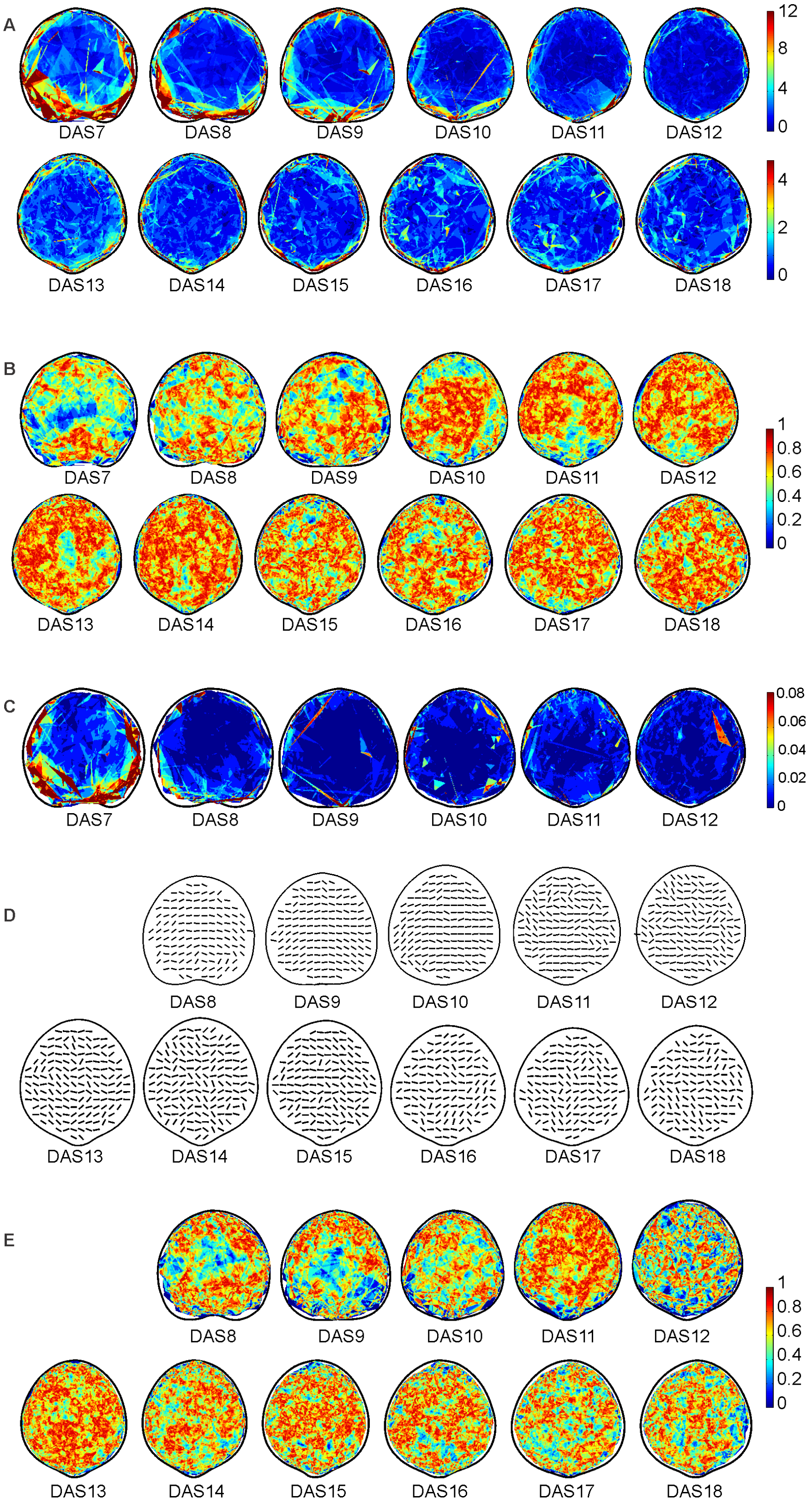

A
Scaling factor p
2.5
2.0
1.5
1.0
8
10
12
14
16
18
DAS
B
Scaling factor q
2.5
2.0
1.5
1.0
Tip
Base
8
10
12
14
16
18
DAS
C
p center-to-side ratio
1.3
1.2
1.1
1.0
0.9
8
10
12
14
16
18
DAS
D
q center-to-side ratio
1.3
1.2
1.1
1.0
0.9
Center
Side
8
10
12
14
16
18
DAS
E
DAS7
DAS8
DAS9
DAS10
DAS11
DAS12
DAS13
DAS14
DAS15
DAS16
DAS17
DAS18
10
8
6
4
2
0
F
DAS7
DAS8
DAS9
DAS10
DAS11
DAS12
DAS13
DAS14
DAS15
DAS16
DAS17
DAS18
10
8
6
4
2
0

Figure S4, related to Figure 6. Analyses of spatiotemporal patterns and error of *p* and *q*, the scaling factors along the maximal and minimal direction of growth, respectively.

(A-B) To illustrate patterns in the scaling factors *p* and *q* at the leaf base and tip over time, we compute and plot the average of the values in the bottom and top quarter (as shown in the diagram) of the mean spatial map of *p* (A) and *q* (B) for each DAS.

(C-D) We assess the gradient shapes by comparing the values of *p* (C) or *q* (D) in a small window at the center to those in a small window at the same lateral position at the side of the spatial map of each sample (as illustrated in the diagram). If the gradient has a downward curving shape, the values in the center will be higher than those at the sides, and the center-to-side ratio will thus be greater than one, with more strongly curving gradients having higher center-to-side ratios, and vice versa. N=12-20. Error bars represent the 95% confidence interval.

(E-F) Spatial maps of the relative standard error (RSE), expressed as a percentage, of the mean maps of *p* (E) and *q* (F). RSE values are plotted by colour according to the colour scales on the right.



A
DAS7
DAS8
DAS9
DAS10
DAS11
DAS12
12
8
4
0
DAS13
DAS14
DAS15
DAS16
DAS17
DAS18
4
2
0
B
DAS7
DAS8
DAS9
DAS10
DAS11
DAS12
1
0.8
0.6
0.4
0.2
0
DAS13
DAS14
DAS15
DAS16
DAS17
DAS18
C
DAS7
DAS8
DAS9
DAS10
DAS11
DAS12
0.08
0.06
0.04
0.02
0
D
DAS8
DAS9
DAS10
DAS11
DAS12
DAS13
DAS14
DAS15
DAS16
DAS17
DAS18
E
DAS8
DAS9
DAS10
DAS11
DAS12
1
0.8
0.6
0.4
0.2
0
DAS13
DAS14
DAS15
DAS16
DAS17
DAS18

Figure S5, related to Figure 7. Error and variance maps of anisotropy and growth direction, and maps of the rotation of growth direction within the tissue over time with associated error maps.

(A) Spatial maps of the relative standard error (RSE) of the mean maps of anisotropy, where RSE values are expressed as a percentage and plotted according to the colour scale on the right.

(B-C) Circular variance maps of the mean direction of growth (B) and angles of tissue rotation (C), where the circular variance values are plotted according to the colour scales on the right. Values closer to zero represent low variance and values closer to 1 represent high variance.

(D) To assess if or how the maximal direction of growth rotates within the tissue over time, we compare the direction of growth within a piece of tissue at two successive periods of growth while taking into account changes in the orientation of the tissue itself. We calculate this by $\theta_{t2 \text{ to } t3} - \theta_{t1 \text{ to } t2} - \Psi_{t1 \text{ to } t2}$, where $\theta_{t2 \text{ to } t3}$ is the direction of maximal growth from time *t2* to *t3*, $\theta_{t2 \text{ to } t3}$ is is the direction of maximal growth from time *t1* to *t2*, and $\Psi_{t1 \text{ to } t2}$ is the rotation of the tissue from time *t1* to *t2.* Thus, in the maps shown, the lines represent how the maximal direction of growth rotates within the tissue in that area compared to growth over the previous time period. A horizontal line indicates zero rotation, i.e. that the maximal direction of growth remains fixed within the tissue.

(E) Circular variance maps of the mean angles of rotation of the direction of growth (D), where values closer to zero represent low variance and values closer to 1 represent high variance. Circular variance values are plotted by colour according to the colour scales on the right.

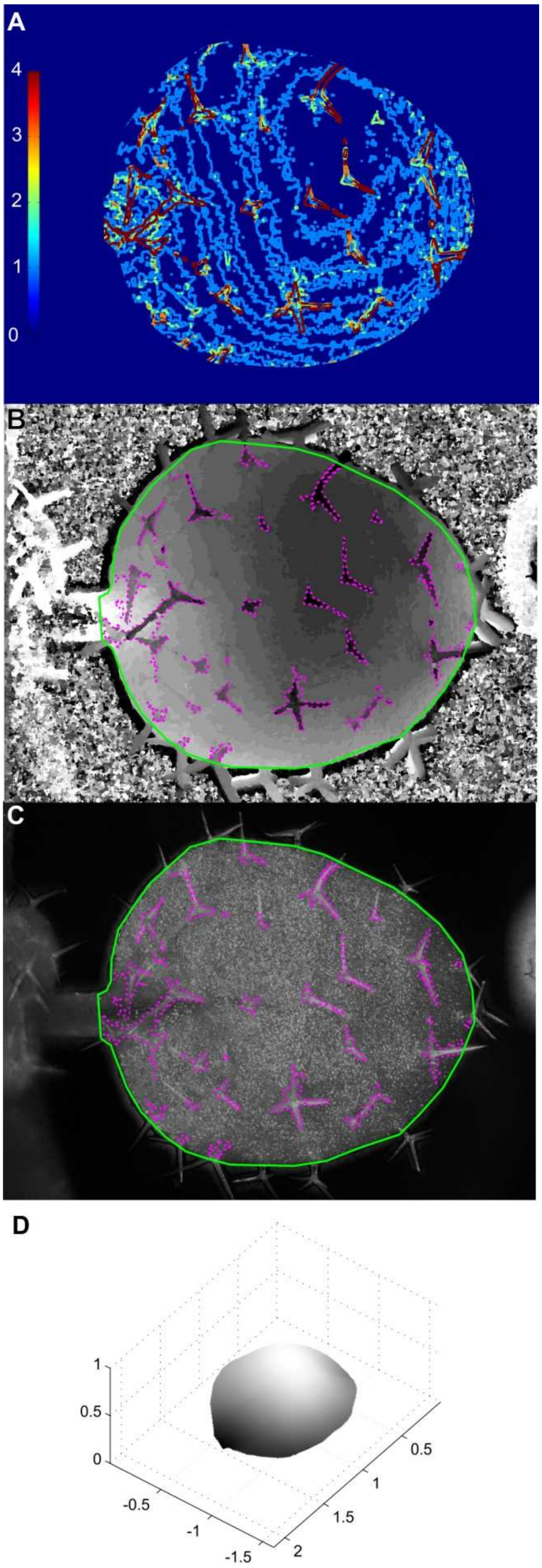


Figure S6, related to Figure 1. Digitally cropping leaf trichomes from wild type *Arabidopsis thaliana* ecotype Columbia.

(A) We first convert the depth map from pixel values to step numbers, by dividing by the difference in pixel values between shades of grey in the depth map. Next we apply the *rangefilt* function from the Image Processing Toolbox, with a window size of 25. This returns a map of the range of step numbers in a window of 25x25 pixels. Aside from trichomes, the leaf surface is smooth, and at no point on a glabrous leaf does a neighboring area of the leaf change in height by more than one level. Trichomes, however, jut out from the surface steeply, and will be several steps above the leaf surface in that area. Therefore, areas where the range of step numbers is greater than 1 correspond to the boundary between the leaf surface and trichomes.

(B-C) We create a new binary image from these areas where the range is greater than 1, and then apply a similar series of steps as in the extraction of the leaf outline: we apply *imdilate* to dilate these areas slightly in case there are any unclosed trichome boundaries, *imfill* to fill in each enclosed area, *imerode* to shrink the areas back down slightly (to offset the previous dilation), then *bwmorph* with the ‘remove’ option to leave only the outlines of each area. The *bwconncomp* function is applied to identify each outline as a separate object, and the *bwtraceboundary* function is applied to extract the x,y-coordinates of each object. These outlines are then plotted in pink on the depth map (B) and multifocus (C) image for the user to verify (with the leaf outline plotted in green). There is an option for the user to add more areas manually if needed. Z-values in the encircled trichome areas are not used when reconstructing the leaf surface in 3D – the program will interpolate the height in these areas based on the height of the surrounding surface.

(D) The 3D reconstruction of the leaf surface after trichomes have been deleted (and the height values are smoothed, as explained in the Supplemental Materials and Methods) is shown. Axis units are in mm.

DAS	Test statistic	RG	*p*	*q*
7	Z	-4.633[a]	-4.165[a]	-6.838[a]
	Asymp. Sig. (2-tailed)	.000	.000	.000
8	Z	-5.147[a]	-4.589[a]	-6.713[a]
	Asymp. Sig. (2-tailed)	.000	.000	.000
9	Z	-5.470[a]	-5.376[a]	-7.129[a]
	Asymp. Sig. (2-tailed)	.000	.000	.000
10	Z	-2.544[a]	-.631[a]	-6.204[a]
	Asymp. Sig. (2-tailed)	.011	.528	.000
11	Z	-1.791[b]	-2.770[b]	-2.342[b]
	Asymp. Sig. (2-tailed)	.073	.006	.019
12	Z	-.559[b]	-1.059[b]	-.789[b]
	Asymp. Sig. (2-tailed)	.576	.290	.430
13	Z	-3.297[b]	-5.679[b]	-3.667[b]
	Asymp. Sig. (2-tailed)	.001	.000	.000
14	Z	-.614[b]	-1.611[b]	-1.113[b]
	Asymp. Sig. (2-tailed)	.539	.107	.266
15	Z	-.366[a]	-.906[a]	-2.545[a]
	Asymp. Sig. (2-tailed)	.714	.365	.011
16	Z	-3.115[b]	-2.961[b]	-2.662[b]
	Asymp. Sig. (2-tailed)	.002	.003	.008
17	Z	-.749[b]	-.640[b]	-1.600[b]
	Asymp. Sig. (2-tailed)	.454	.522	.110
18	Z	-1.619[b]	-3.432[b]	-1.210[b]
	Asymp. Sig. (2-tailed)	.105	.001	.226

a. Based on negative ranks.

b. Based on positive ranks.

Table S1, related to Figure 5. Testing the gradient shapes of RG, *p*, and *q* at each time point.

To statistically assess the gradient shapes observed for RG, *p* and *q*, we compared the values in a small window at the center to those in a small window at the same longitudinal position at the sides of each single sample leaf spatial map (refer to diagram in Figure S4C-D), with a Wilcoxon signed-rank test. If the observed curvature of the gradients are significant, the difference between the values at the center will be significantly different from the values at the sides. Asymp. Sig. refers to asymptotic significance.

SUPPLEMENTAL MATERIALS AND METHODS

More details and techniques on particle application

If the droplet of particle solution does not fully cover the leaf surface, this can be resolved by adding a small amount of the more concentrated solution to the droplet, and then overlaying it with a tiny piece of tissue, which will break the spherical shape of the droplet and force the solution to spread out across the leaf; the tissue is removed after it has dried (approximately 60 minutes).

We generally apply the particles to both of the first two rosette leaves of the chosen plants, and then select the leaf with the best particle coverage for further imaging. If the dispersion or concentration of particles is not ideal, the particles can be pushed around on the leaf surface with a small piece of wet tissue to improve dispersion, blotted with the tip of a moist tissue to remove some particles in concentated areas, or blotted with the tip of a tissue dipped in the more concentrated solution to add particles to sparse areas. However, because mechanical stimulation is known to affect growth (Chehab et al., 2009), these manipulations are kept to a minimum, and if good particle coverage is not easily achieved, the plant is discarded.

Details on settings for microscopy and image acquisition

There are several settings on the microscope and camera (Leica DFC350) that can be adjusted to optimize the images obtained. As over-exposure to light could potentially alter plant growth, we are always conscious of choosing settings that also minimize the intensity and duration of light exposure. We use the lowest levels of brightfield light intensity, adjusting the brightness as needed, from 10 to 30, and the lowest level of fluorescent light intensity, adjusting the brightness from 6.3 to 100%. The 2.0x objective lens is used for smaller leaves (up to around DAS15) and the 1.0x lens is used for the larger leaves.

The z-stack is acquired by taking a series of images at regular vertical intervals within the focal range of the specimen, which is set manually by setting the bottom of the range to where the lowest point of the leaf is just out of focus and the top end to where the highest point of the leaf is just out of focus. The regular vertical interval size is referred to as the step size, and is optimal when it is small enough to capture the leaf curvature and shape in 3D (which is important for generating a clear montage image and accurate depth map) without being redundant and causing unneccessary longer exposure of the plant under the microscope lights. A visual assessment of the depth map quality is made from the confidence map, in which grayscale values ranging from black to white represent 0-100% accuracy (Figure 1C). The step size can be set automatically by the Leica® software's "Optimize step size" option, which adjusts with the 2D size (zoom) of the specimen, or it can be entered manually to obtain the desired vertical resolution. For fluorescence images (for which the depth map is not used), the step size can be increased to shorten the leaves' exposure to the fluorescent light, since we do not use the depth map of the fluorescence stack.

To visualize the fluorescent yellow particles we use a GFP filtercube (excitation filter BP 470/40, dichromatic mirror 500, suppression filter BP 525/50), with the intensity of the fluorescent light source kept as low as possible. The camera settings for fluorescent images need to be determined on a case-by-case basis, as the best settings depend on the density of particle coverage, and the apparent brightness of the particles. The latter depends on the size of the particles and the relative size of the leaf; for example, a 29.6µm particle appears quite large and bright on a leaf at DAS15, but will appear much smaller and fainter (until the camera settings are adjusted) at DAS21 when the leaf is comparatively much larger. We find the lowest levels of fluorescent light intensity to be sufficient, with exposure times ranging from 700-900ms, and gamma correction and gain usually set around 0.55 and 3.2x respectively. The same camera settings can usually be used for the brightfield images as well.

Computational details of the particle tracking algorithm

For each particle on a leaf at time *t1*, the pattern-matching algorithm computes and records a "neighbour pattern" based on the *n* closest neighbouring particles; the default number of neighbours used in the pattern is three, but this can be changed by the user if desired. A 2D coordinate system is centered on the particle in question, and we compute the polar coordinates (radius, θ) of the *n* neighbouring particles within that coordinate system, through the Matlab *cart2pol* function (built-in Matlab functions are cited in italics throughout). We use the radius to compute the relative distance of each neighbour from the particle in question as the radius divided by the 2D length of the leaf. θ gives the angle of the neighbour relative to the the particle in question. We store the relative distances of the neighbours in one matrix (D_{t1}) and their orientations in another (O_{t1}); each matrix has the dimensions *p* by *n*, where *p* is the number of particles on the leaf, and *n* is the number of neighbouring particles for which we store data. Each row *p* of D_{t1} contains the relative distances of the *n* closest neighbours to particle *p*, and the corresponding indices of O_{t1} contain their orientations. Data for neighbours is always stored by increasing distance from left to right; i.e. data for the closest neighbour is in the 1st column, data for the second closest neighbour is in the 2nd column, etc.

This is repeated for all particles at *t2*. Now the neighbour pattern of each particle at time *t1* is compared to the neighbour pattern of each of the particles at *t2*, based on the ratio of the relative distances and difference in orientation of the closest neighbours. Because growth is not uniform, we do not expect the pattern to remain exactly the same and we allow for some discrepancy. The default discrepancy permitted is 30% for the distance ratio and 0.15 radians for the angle difference, but these can be adjusted by the user if need be. When the neighbour pattern of a particle at *t1* (*p1)* is sufficiently similar to the neighbour pattern of a particle in at *t2 (p2)*, it is recorded as a potential match. I.e.

```
FOR $p1$ = 1 to the number of particles on the leaf at $t1$
    FOR $p2$ = 1 to the number of particles on the leaf at $t2$
        FOR $n$ = 1 to the number of neighbours used
            $dD(n)=D_{t1}(p1,n) / D_{t2}(p2,n)$
            $dO(n)=abs(O_{t1}(p1,n)-O_{t2}(p2,n))$
        EndFOR
        IF $\max(dD) < 0.3$ AND  $\max(dO) < 0.15$
            Particle $p1$ and $p2$ are recorded as a potential match
        EndIF
    EndFOR
EndFOR
```

Additional checks at the end of the program will remove any matches where more than one potential match was identified, and otherwise saves the best match.

It is usually possible to match an estimated 80-90% of the particles with this algorithm, with 100% accuracy most of the time. However, matches should be verified by the user, and if an insufficient number of matches were found, or an erroneous match found, there is an option to add or delete particles manually.

Once most of the particles have been matched, their change in coordinates between the successive time points can be used to warp all of the *t1* coordinates to *t2* using piecewise transformations. Such warping is done by grouping matched particles into vertices of non-intersecting triangles using Matlab's *delaunay* triangulation function. For each of these triangles, an affine transformation matrix is calculated based on the change in the vertex coordinates from time *t1* to *t2* using the *cp2tform* function, and is applied to the *t1* particles making up the triangle and any that lie within it via the *tformfwd* function. This is repeated for all triangles, and a second-order polynomial function, derived from all of the matched points, is applied to remaining particles around the perimeter that do not lie within any of the triangles (again using the *cp2tform* and *tformfwd* functions).

The result is perfect alignment of the matched particles and usually a very good alignment of the remaining un-matched particles on the leaf at *t2*. The remaining particles can now be matched according to the alignment, by finding the warped *t1* particle which is the closest to each *t2* particle. If necessary, additional matches can be manually added at this point in areas that are crowded or not well aligned. There are again a few checks in place that can be adjusted by the user if desired to increase the number of matches or increase the accuracy, including the maximum distance allowed between particles in order for them to be considered a match, and a minimum distance allowed between the top two matches (i.e. if there are potentially two matches for a particle, no match will be recorded). There is also an option to consider a match when it is not within the minimum distance but if there are no other possible matches nearby.

Digitally editing and modifying the leaf surface

The depth map groups the image into regions that are most in focus at each z-step but in reality we know the curved leaf spans in-between the focal planes. To convert the depth map from a surface of flat steps to a more realistic continuously curved surface, it is “smoothed” using a square averaging window with dimensions of 50 pixels (on the 1041x1393 image) using the *fspecial* and *imfilter* functions. The leaf surface can now be described by a 3D grid, from a set of points fitted to the leaf shape with z-coordinates obtained from the smoothed depth map. The 3D leaf surface is then plotted and any aberrent areas can be edited by manually cropping out problem regions on the depth map and interpolating new z-values in those regions, although this procedure is usually only necessary when using leaves with trichomes (see Figure S5).

Mathematical details on the computation of growth strain parameters in 3D

Before the growth parameters for a given triangle can be computed, the 3D triangle coordinates must be rotated so that they can be described with 2D coordinates. The triangle is shifted so that one vertex lies at the origin, and the normal vector to a plane fit through the triangle is found using the cross product of two of the triangle points. The projection of the angle of this vector in the x-y plane (α) is obtained, and is essentially equal to the 2D angle between the normal and the positive x-axis. The triangle is rotated clockwise around the z-axis by the angle α, bringing the projection of the normal in the x-y plane in line with the positive x-axis. Consequently this puts the intersection of the triangle and the x-y plane along the y axis. The angle between the normal and a vector along the z-axis (β) is now calculated using the dot product. The projection of the normal is in line with the x-axis, so if the triangle is now rotated around the y-axis by the angle β, the normal will become perfectly aligned with the z-axis in 3D. Consequently the triangle will now lie in the x-y plane. This is repeated for both the time *t1* and *t2* triangles.

Now growth from the two flattened triangles can be calculated according to the singular value decomposition, described by Goodall and Green (1986), of a transformation matrix, *Tr*, that mathematically describes the transformation of a triangle’s landmarks from *t1* to *t2*. This will produce the scaling factors (p and q), the direction of maximal growth (θ_f), and the triangle rotation (ψ_f), of the flattened triangles.

Since the angles obtained are computed after the triangles are rotated on to the x-y plane, the actual direction of growth and rotation of tissue must be adjusted to account for this. To compute the overall rotation of the triangle, we apply a total transformation matrix (the *t1* flattening rotations, the rotations and scaling of growth, and the inverse of the *t2* flattening rotations) to a unit vector laying along the positive x-axis. The resulting position of the vector gives the final angle of rotation the tissue experienced during growth, as measured from the positive x-axis. To align the direction of growth with the position of the triangle within the 3D leaf, we apply the inverse of the *t1* flattening rotations to a unit vector in the direction of (θ_f).

The resulting angles for the direction of growth and rotation of tissue have two components: the angle in the x-y plane, and the angle from the x-y plane. For the purposes of making deductions about growth signals, the relevant angle is the angle within the plane of the tissue. However this is complicated to visualize, since publications only allow for 2D images, or projections of 3D images, and since the leaf is fairly flat, we decided it was appropriate to just show the 2D projections of the angles.

Shifting data using leaf area as an indicator of developmental stage

Days after sowing may not necessarily reflect the growth stage of a leaf, as not all plants germinate at the same time. In order to group the data as accurately as possible, we sowed approximately 100 seeds, and of those that germinated, selected 24 which had fully opened cotyledons at DAS4 and first leaves of approximately the same size at DAS7, and randomly divided these into two sets. One half was followed from DAS7 to DAS12, and the second was followed from DAS12 to DAS19. Since the imaging did not affect the leaf growth, we are able to combine the two data sets to create a continuous standard growth curveof a 2D projection of the leaf areas.

As leaves from the first data set grew, particles in fast growing areas became further apart, leading to less particle coverage at the base of DAS10-12 leaves. We therefore sowed an additional set of plants, which were imaged from DAS10 to DAS14. Due to variations in germination times, we wrote a program to automatically align each plant to the standard growth curve according to a weighted least squares fit of the leaf areas over time.

SUPPLEMENTAL REFERENCES

Chehab, E.W., Eich, E., and Braam, J. (2009). Thigmomorphogenesis: a complex plant response to mechano-stimulation. J Exp Bot. 60, 43-56.