CROPS AS TIME-INVARIANT KEYPOINTS

M. BOS [*]

DR. N.V. BUDKO [†]

IR. E. VERHOEFF [‡]

JULY 2020

CONTENTS

ABSTRACT

In this paper, a method is proposed to automatically correct misalignment of orthophotos in time-series caused by an inaccurate geotransform. The proposed method relies on common literature concepts such as keypoint identification, keypoint matching, and model fitting using random sample consensus (RANSAC). Traditional keypoint identification methods such as the scale invariant feature transform (SIFT) are not suited for this problem as no real scale- or rotation-invariance is required, instead, time-invariance is required. To achieve this, crops are suggested as keypoints, and two different keypoint descriptors are put forth. The first descriptor is based on the shape and size of the crops, while the alternative descriptor is based on the planting pattern of crops. The method, and both descriptors, generate promising results for certain scenario's. However, in later growth stages performance drops significantly as the identification of crops -required beforehand-becomes troublesome due to dense growth.

[*] *M.Sc. Student Applied Mathematics, University of Technology Delft, The Netherlands*

[†] *Associate Professor Numerical Analysis, University of Technology Delft, The Netherlands*

[‡] *Co-Founder & Head Technical Operations, VanBoven Drones B.V., The Netherlands*

1 INTRODUCTION

In the agricultural sector, crop monitoring via drones is an increasingly used technique. Often times, one is interested in a time-series, and in doing change detection between instances of this time-series, for example, establishing the rate at which a crop has grown. Before doing such analysis, however, image stitching using a photogrammetry pipeline is required. Using common software such as *Agisoft Metashape* or *OpenDroneMap*, one may combine the separate images acquired by the drone into what is referred to as an 'orthophoto'. An orthophoto is geometrically correct and may be used as a map in the sense that one can directly infer distances from it. Furthermore, it has a 'geotransformation' attached to it, which assigns a latitude and longitude to each pixel in the orthophoto.

In this attached geotransformation one is presented with the problem this paper attempts to address. The geotransformation, being a result of the GPS built into the camera of the drone, is accurate in the order of ten meters.[1] It is clear how this is problematic given the scope of change detection. Correction can be performed manually by creating so-called 'ground control points', which are corresponding points between two orthophotos, after which a transformation may be applied. This paper will attempt to provide a framework to automate the creation of such ground control points by using an approach comparable to what is used in common photogrammetry pipelines for stitching images into an orthophoto.

[1] Indeed, the problem may also be addressed by more accurate GPS positioning, however, currently, this is a problematic path to take for two reasons. The cost associated with such a GPS systems is rather high, while the payload is also drastically larger, creating various new problems such as the requirement for larger drones and a higher fuel consumption.

2 CREATION OF ORTHOPHOTOS

When processing separate images in order to create an orthophoto, see figure 3, a three-dimensional reconstruction of the scene is required first. Crucial for creating this three-dimensional reconstruction, given a set of two-dimensional images, is the *essential matrix* or E, introduced originally by Longuet-Higgins (1981).

2.1 Essential Matrix

The essential matrix follows directly from the coplanarity condition. Suppose we have two cameras, camera A and B, in a three-dimensional space, see figure 1. Both cameras have their own coordinate system, (X_A, Y_A, Z_A) and (X_B, Y_B, Z_B), respectively. As a convention, the Z-direction will always coincide with the direction in which the camera is pointed. The translation between the cameras is given by the vector $\mathbf{t}$, which is a vector in the coordinate system of camera A, pointing the origin of that of camera B.

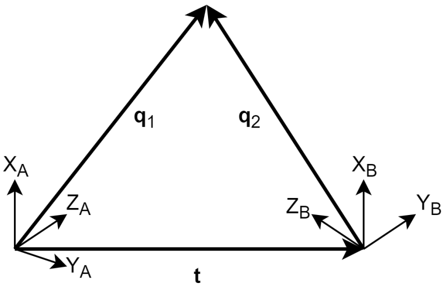


Figure 1: Two cameras and their three-dimensional coordinate systems, observing a single point in space that is seen by both cameras, $\mathbf{q}_1$ and $\mathbf{q}_2$.

Now, suppose both cameras observe some point in space. The position of this point with respect to camera A may be denoted by the vector $\mathbf{q}_1$, while the position with respect to camera B is given by $\mathbf{q}_2$. Note that $\mathbf{t}$, $\mathbf{q}_1$ and $\mathbf{q}_2$ are coplanar. However, the coplanarity condition cannot be applied directly, as $\mathbf{q}_2$ is a vector in (X_B, Y_B, Z_B) while the other vectors are given in the coordinate system of camera A. Let R denote the rotation between the two coordinate systems, consequently, coplanarity should hold for $\mathbf{t}$, $\mathbf{q}_1$ and $R\mathbf{q}_2$.

If three vectors are coplanar, taking the cross product between two of them produces a vector orthogonal to the plane. Consequently, taking the dot product with the third vector should then always yield zero. Thus, by coplanarity:

$$\mathbf{q}_1 \cdot (\mathbf{t} \times R\mathbf{q}_2) = 0. \tag{1}$$

The cross-product with the three-dimensional translation vector $\mathbf{t}$ is equivalent to multiplication with the 3×3 matrix representation $[\mathbf{t}]_\times$, defined as:

$$[\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}. \tag{2}$$

Since R is a 3×3 rotation matrix, we may define the essential matrix as $E = R[\mathbf{t}]_\times$, again a 3×3 matrix, resulting in:

$$\mathbf{q}_1^T E \mathbf{q}_2 = 0. \tag{3}$$

Given this definition of the essential matrix, one may wonder how it relates to two-dimensional image coordinates. Consider one camera and the corresponding image plane, see Figure 2.

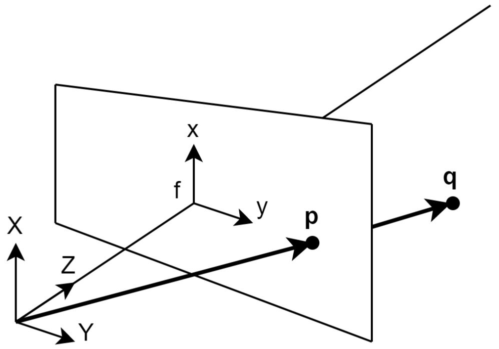


Figure 2: One camera and the corresponding image plane a focal lenght f in front of the camera, observing a single point that is given by **q** and forms projection **p** on the image plane.

Note that the image plane is actually a focal length f behind the camera and inverted, but for geometric reasons it may as well be placed normally in front of the camera. Consequently, in the pinhole camera model, the relation between image and three-dimensional coordinates is given by:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix}. \tag{4}$$

Without loss of generality, the scene may be scaled by setting $f = 1$, resulting in homogeneous image coordinates:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \tag{5}$$

As a result, an observed point $\mathbf{q}$ in three dimensional space is projected on the image plane in $\mathbf{p}$, given by $\mathbf{p} = (1/q^Z)\mathbf{q}$, where q^Z denotes the Z-component of $\mathbf{q}$. Note that this implies $\mathbf{p}$ is a three-dimensional vector, referencing a two-dimensional point in the image plane.

Consider again figure 1 and equation (3). Given the found relation between three- and two-dimensional coordinates, it can be shown that the essential matrix also relates image coordinates:

$$\mathbf{q}_1^T E \mathbf{q}_2 = 0, \tag{6}$$

$$\left(\frac{1}{q_1^Z}\right)\left(\frac{1}{q_2^Z}\right) \mathbf{q}_1^T E \mathbf{q}_2 = 0, \tag{7}$$

$$\mathbf{p}_1^T E \mathbf{p}_2 = 0. \tag{8}$$

Since the essential matrix also relates image coordinates, correspondences between two images may be used to infer the essential matrix and thus translation and rotation between the cameras.

Given $\mathbf{p}_1$ in image A that corresponds to $\mathbf{p}_2$ in image B, (8) must hold. Rewriting this gives:

$$\begin{bmatrix} p_1^x p_2^x & p_1^x p_2^y & p_1^x & p_1^y p_2^x & p_1^y p_2^y & p_1^y & p_2^x & p_2^y & 1 \\ & & & & \vdots & & & & \end{bmatrix} \begin{bmatrix} E_{11} \\ E_{12} \\ \vdots \\ E_{33} \end{bmatrix} = 0. \tag{9}$$

Here, the top row contains the known x- and y-components of the identified points, while the column vector contains elements of the unknown essential matrix. If more

corresponding points are identified, they may be added as extra rows, arriving at the linear system $A\boldsymbol{e} = 0$, with $\boldsymbol{e} = \langle E_{11}, ..., E_{33} \rangle^T$.
Consider (8), clearly, it would still hold for $\hat{E} = kE$ with any scalar k. In order to avoid this scaling problem as well as the trivial solution to $A\boldsymbol{e} = 0$, the constraint $\|\boldsymbol{e}\| = 1$ is added (Hartley, 1997). Now, eight linear independent rows determine an unique nontrivial solution. Since then $A \in \mathbb{R}^{8\times9}$ and the rows are linearly independent, one has $\text{rank}(A) = 8$. Consequently, A has eight non-zero singular values.
Consider the singular value decomposition of A as:

$$A = U\Sigma V^T, \tag{10}$$

where $U \in \mathbb{R}^{8\times8}$, $\Sigma \in \mathbb{R}^{8\times9}$, and $V \in \mathbb{R}^{9\times9}$. The columns of V contain the right-singular vectors of A. Since A has eight non-zero singular vectors, exactly one right singular vector in V will correspond to a zero singular vector. By definition, this vector spans the null space of A. Normalising this vector provides an $\boldsymbol{e}$ that satisfies $A\boldsymbol{e} = 0$ and $\|\boldsymbol{e}\| = 1$. See also Hartley (1997) for details on deriving E from eight correspondences.

The fact that eight rows are required, and thus eight point correspondences, leads to the so-called 'eight-point problem', originally introduced as well by Longuet-Higgins (1981). Moreover, it is possible, though not as straight-forward, to use as little as five correspondences, see Fathian et al. (2018).
Solving the eight-point problem, that is, finding at least eight corresponding points between two images, allows one to determine the essential matrix. Consequently, translation and rotation may be inferred, see Nistér (2004). Given translation and rotation, points may be triangulated, creating one combined three-dimensional reconstruction of the scene. Finally, this pointcloud may be projected onto a flat surface, creating the desired orthophoto, see figure 3.

2.2 Point Correspondences

In order to solve the eight-point problem itself, a set of point correspondences between images is required first. This set of correspondences can be retrieved by use of keypoints in images. Finding correspondences by keypoints can be divided in number of subsequent steps: determining keypoint locations, associating keypoint descriptors with each location, using descriptors to match keypoints in different images, and finally, model-fitting.

2.2.1 *Keypoint Location and Descriptor*

Broadly speaking, keypoint identification consists of two steps; identifying locations and describing these. There exist various algorithms for this purpose, including SIFT, SURF, KAZE, ORB, BRISK, and others (Li et al., 2015). Presumably, the Scale Invariant Feature Transform (SIFT) is the best known algorithm amongst these, introduced by Lowe (1999). SIFT finds keypoint locations, $\mathbf{p}_i$, by analysing the Gaussian scale-space of an image, thus creating scale invariance. Subsequently, each keypoint is described by a corresponding descriptor vector, $\mathbf{f}_i$, in a way that is rotation invariant, and illumination invariant to some degree. See Rey-Otero and Delbracio (2014) for a detailed explanation of SIFT.

2.2.2 *Keypoint Matching*

Next, keypoints should be matched such that correspondences between images are created. This is achieved by matching the descriptors to their nearest neighbours. Given image A that has keypoints i with $i \in S^A$ and image B with keypoints that have index $i \in S^B$. Each keypoint i has a location $\mathbf{p}_i$ and a descriptor $\mathbf{f}_i$ assigned by e.g. SIFT. For some keypoint $i \in S^A$, a candidate match ι is defined as the nearest neighbour in the descriptor space:

$$\iota = \arg\min_{\tau \in S^B} \|\mathbf{f}_i - \mathbf{f}_\tau\| \,. \tag{11}$$

As suggested by Lowe (2004), this candidate will be accepted if and only if:

$$\|\mathbf{f}_i - \mathbf{f}_\iota\| < C \min_{\tau \in S^B \setminus \{\iota\}} \|\mathbf{f}_i - \mathbf{f}_\tau\| \,, \tag{12}$$

where $C \leqslant 1$ is a constant. In other words, ι will be accepted as match to i if there is no close runner-up match, which is checked by considering the distance to the second-nearest neighbour in the descriptor space.

2.2.3 *Model Fitting*

When keypoints are matched between images, the model, in this case the essential matrix E, should be fit to the data. Commonly used techniques such as least-squares are ill-suited for this problem, since data does not always consist of 'model plus Gaussian noise' terms and outliers may have disproportionally large errors. Hence, the RANdom SAmple Consensus (RANSAC) algorithm is commonly used for model fitting (Fischler and Bolles, 1981).

For a number of iterations, a random sample of size eight -recall that eight correspondences were required in (9)- is drawn from the set of all matches. Corresponding to this sample we find a certain candidate essential matrix $\tilde{E}$ by solving the system. This $\tilde{E}$ is tested against all data by checking how many matches are 'inliers', i.e., satisfy:

$$\mathbf{p}_i^T \tilde{E} \mathbf{p}_\iota < \varepsilon, \tag{13}$$

for some threshold error ε. The $\tilde{E}$ corresponding to the random sample that produced the largest number of inliers after all iterations is saved as the true essential matrix E.

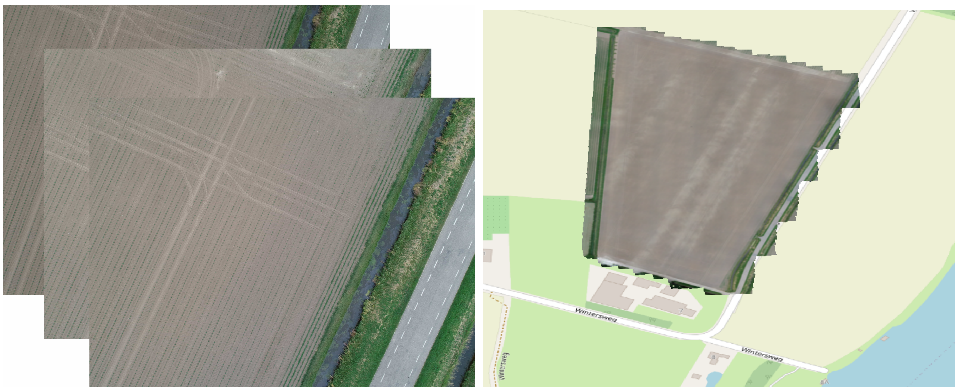


Figure 3: The three images on the left, which were taken by the drone, are used -amongst others- to create the orthophoto on the right. Notice the images depict a road and a ditch, which is recognisable again on the right-hand side of the orthophoto. Furthermore, notice also that the orthophoto is depicted overlaid on the *OpenStreetMap* [2] reference map, which is possible due to the attached geotransform.

[2] https://www.openstreetmap.org

3 TIME ALIGNMENT OF ORTHOPHOTOS

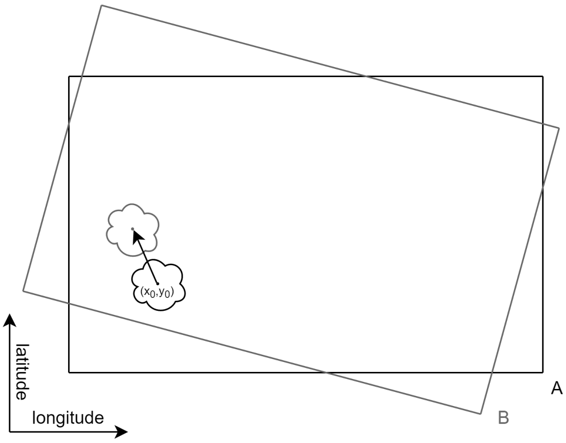


Figure 4: Orthophoto A and B taken at different dates misaligned in a common reference frame. The point $\langle x_0, y_0 \rangle^A$ refers to some crop centroid, however, due to the misalignment, $\langle x_0, y_0 \rangle^B$ does not refer to this centroid anymore.

The geotransform attached to orthophotos that places them in a common reference frame is correct only up to roughly ten meters, causing misalignment between orthophotos. Consequently, for two orthophotos A and B taken at different dates, $\langle x_0, y_0 \rangle^A$ may not refer to the same location as $\langle x_0, y_0 \rangle^B$, see figure 4. It is observed that the error may mostly be characterised by translation, as well as rotation in the two-dimensional plane, see figure 11 for an explanation. Possibly, by accumulation of error in the photogrammetry pipeline, minor local deformations may be found as well. It is thus assumed that the error between orthophoto A and B may be characterised as:

$$\mathbf{x}^A = \mathbf{t} + R\mathbf{x}^B + \boldsymbol{w}(\mathbf{x}^B), \tag{14}$$

where $\mathbf{t}$ is a two-dimensional translation vector, R is a 2×2 rotation matrix, and $\boldsymbol{w}$ is some unknown vector field. If $\mathbf{t}$, R, and $\boldsymbol{w}$ are known, orthophoto B may be corrected by $\mathbf{x}^{B_c} = \mathbf{t} + R\mathbf{x}^B + \boldsymbol{w}(\mathbf{x}^B)$, such that now, $\langle x_0, y_0 \rangle^A$ refers to the same locations as $\langle x_0, y_0 \rangle^{B_c}$.

The unknown error -characterised by $\mathbf{t}$, R, and $\boldsymbol{w}$- may be found by finding correspondences between orthophoto A and B. In the context of aerial imaging, these correspondences are usually called ground control points and are created by identifying man-made markers in the scene. As this method is labour intensive, it may prove advantageous to use a method comparable to the method presented in literature for finding point correspondences between pairs of images.

However, traditional methods, such as keypoint identification by the scale invariant feature transform (SIFT), are not suited for the problem as no real scale or rotation invariance is required. Instead, time-invariant keypoints are required that are immune to the changes that occur within the time-series such as growth of vegetation, change of lighting, or change of weather. Crops are put forth as a candidate for the creation of such keypoints since they are guaranteed to be present in the orthophoto, but also the main interest within the orthophoto, making it advantageous to have correspondences in close proximity to them.

3.1 Modelling Growth

In order to facilitate time-invariance, growth of crops must be modelled such that it can be accounted for in creating correspondences. Let a crop contour, in nadir view, at some time t be described by a parametrized vector function using polar coordinates:

$$\mathbf{r}(\xi, t) = r(\xi, t)\mathbf{n}(\xi) + \mathbf{s}, \qquad \xi \in [0, 2\pi], \ \mathbf{n}(\xi) = \langle \cos\xi, \sin\xi \rangle, \tag{15}$$

where $\mathbf{s}$ denotes the position of the stem of the crop, which is fixed and thus independent of t, see figure 5.

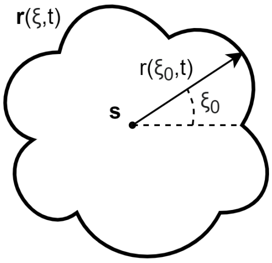


Figure 5: Crop contour $\mathbf{r}(\xi, t)$ at some time t, see (15). The contour is described in polar coordinates by the radius $r(\xi, t)$ from the fixed position $\mathbf{s}$ to the edge of the crop for $\xi \in [0, 2\pi]$.

Given that crops exhibit periods of exponential growth, while tending to a finite circular shape, a logistical growth model was assumed:

$$\frac{\partial r}{\partial t} = \alpha r \left(1 - \frac{r}{R}\right), \tag{16}$$

where R represent the maximum achievable radius of a crop. The expansion rate, that is, the speed at which a crop grows, is given by the constant α. Integrating (16) with respect to t, an expression for $r(\xi, t)$ may be found as:

$$r(\xi, t) = \frac{R}{1 + \frac{R - r(\xi, t_0)}{r(\xi, t_0)} e^{-\alpha t}}. \tag{17}$$

Now suppose that for some t_0 we have that $r(\xi_1, t_0) > r(\xi_2, t_0)$. It can be shown that then it must also hold that $r(\xi_1, t) > r(\xi_2, t)$ for all $t > t_0$:

$$r(\xi_1, t_0) > r(\xi_2, t_0), \tag{18}$$

$$\frac{R}{r(\xi_1, t_0)} < \frac{R}{r(\xi_2, t_0)}, \tag{19}$$

$$1 + \frac{R - r(\xi_1, t_0)}{r(\xi_1, t_0)} e^{-\alpha t} < 1 + \frac{R - r(\xi_2, t_0)}{r(\xi_2, t_0)} e^{-\alpha t}, \tag{20}$$

$$\frac{R}{1 + \frac{R - r(\xi_1, t_0)}{r(\xi_1, t_0)} e^{-\alpha t}} > \frac{R}{1 + \frac{R - r(\xi_2, t_0)}{r(\xi_2, t_0)} e^{-\alpha t}}, \tag{21}$$

$$r(\xi_1, t) > r(\xi_2, t), \tag{22}$$

where (19) holds since $R > 0$, (20) holds since $r > 0$ and $e^{-\alpha t} > 0$, and (21) is true again since $R > 0$. Consequently, it is thus shown that if one were to sample radii of a crop at a set of ξ and order the results, the order cannot change over time. Moreover, assuming that for some specimen all crops have the same α and R, it is true that the global order of radii cannot change over time. This observation, based on (22), may be used later to 'create' time-invariance.

3.2 Time-Invariant Correspondences

As presented in literature, correspondences are usually found by determining keypoint locations $\mathbf{p}_i$, describing these in a descriptor $\mathbf{f}_i$ which are then used to match keypoints, and finally, the matched keypoints are filtered using RANSAC and the essential matrix E. In this section an adaptation of this pipeline is presented based on the concept of using crops as keypoints, using the previously introduced crop parametrization and growth model.

3.2.1 *Keypoint Location*

Since each identified crop will be a keypoint, the keypoint location $\mathbf{p}_i$ of keypoint, or crop, i may intuitively be selected as the crop's stem $\mathbf{s}$, see (15) and figure 5. This choice offers attractive properties as it's position cannot change over time. However, the position of the stem is not visible in the airborne orthophoto, thus, instead, the centroid of the foliar mass will be used as a substitute. The location of the centroid is defined as:

$$\mathbf{c}(t) = \frac{1}{A(t)} \iint_{\Omega(t)} \mathbf{r} d\Omega, \tag{23}$$

with $A(t)$ the area of the crop and $\Omega(t)$ the domain defined by $\mathbf{r}(\xi, t)$ in (16). Now, time evolution of the centroid is governed by:

$$\begin{aligned}\mathbf{c}(t) &= \frac{1}{A(t)} \left(\iint_{\Omega(t)} \mathbf{c}(t_0) d\Omega + \iint_{\Omega(t)} \Big(\mathbf{r} - \mathbf{c}(t_0)\Big) d\Omega \right), & (24)\\ &= \mathbf{c}(t_0) + \frac{1}{A(t)} \int_0^{2\pi} \mathbf{n}(\xi) \int_0^{r(\xi,t)} \rho^2 d\rho d\xi, & (25)\\ &= \mathbf{c}(t_0) + \frac{1}{3A(t)} \int_0^{2\pi} r^3(\xi, t)\mathbf{n}(\xi) d\xi. & (26)\end{aligned}$$

Clearly, it must hold that $\int_0^{2\pi} r^3(\xi, t_0)\mathbf{n}(\xi) d\xi = \mathbf{0}$.
Notice that, when growth is purely exponential, one has $\mathbf{c}(t) = \mathbf{c}(t_0)$, since:

$$\mathbf{c}(t) = \mathbf{c}(t_0) + \frac{e^{3\alpha t}}{3A(t)} \int_0^{2\pi} r^3(\xi, t_0)\mathbf{n}(\xi) d\xi = \mathbf{c}(t_0). \tag{27}$$

Additionally, when growth is of logistic nature, as was assumed in (16), but $r(\xi, t_0)$ is perfectly circular, i.e. $r(\xi, t_0) = r_0$ for some constant r_0, one also finds $\mathbf{c}(t) = \mathbf{c}(t_0)$, since by substitution of (17) in (26):

$$\begin{aligned}\mathbf{c}(t) &= \mathbf{c}(t_0) + \frac{R^3}{3A(t)} \int_0^{2\pi} \left(1 + \frac{R - r(\xi, t_0)}{r(\xi, t_0)} e^{-\alpha t}\right)^{-3} \mathbf{n}(\xi) d\xi, & (28)\\ &= \mathbf{c}(t_0) + \frac{R^3}{3A(t)} \int_0^{2\pi} \frac{r^3(\xi, t_0)}{\Big(Re^{-\alpha t} - r(\xi, t_0)e^{-\alpha t} + r(\xi, t_0)\Big)^3} \mathbf{n}(\xi) d\xi, & (29)\\ &= \mathbf{c}(t_0) + \frac{R^3}{3A(t)\Big(Re^{-\alpha t} - r_0 e^{-\alpha t} + r_0\Big)^3} \int_0^{2\pi} r^3(\xi, t_0)\mathbf{n}(\xi) d\xi = \mathbf{c}(t_0). & (30)\end{aligned}$$

On the contrary, when $r(\xi, t_0)$ is irregularly shaped, and growth is of a logistic nature as in (16), one may expect the centroid to drift when comparing with $t > t_0$. In order to investigate the magnitude of this drift, a Taylor series of $r^3(\xi, t)$ around t_0 is substituted in (26):

$$\begin{aligned}
\mathbf{c}(t) = \mathbf{c}(t_0) &+ \frac{1}{3A(t)} \int_0^{2\pi} \left(r^3(\xi, t_0) + \frac{\partial r^3(\xi, t_0)}{\partial t}(t - t_0) + \frac{\partial^2 r^3(\xi, t_0)}{2\partial t^2}(t - t_0)^2 + \dots \right) \mathbf{n}(\xi) d\xi, \\
= \mathbf{c}(t_0) &+ \frac{1}{3A(t)} \int_0^{2\pi} 3r^2(\xi, t_0) \frac{\partial r(\xi, t_0)}{\partial t}(t - t_0) \mathbf{n}(\xi) d\xi \\
&+ \frac{1}{3A(t)} \int_0^{2\pi} \left(\frac{\partial^2 r^3(\xi, t_0)}{2\partial t^2}(t - t_0)^2 + \dots \right) \mathbf{n}(\xi) d\xi, \\
\overset{(I)}{=} \mathbf{c}(t_0) &+ \frac{\alpha(t - t_0)}{A(t)} \int_0^{2\pi} r^3(\xi, t_0) \left(1 - \frac{r(\xi, t_0)}{R} \right) \mathbf{n}(\xi) d\xi \\
&+ \frac{1}{3A(t)} \int_0^{2\pi} \left(\frac{\partial^2 r^3(\xi, t_0)}{2\partial t^2}(t - t_0)^2 + \dots \right) \mathbf{n}(\xi) d\xi, \\
= \mathbf{c}(t_0) &- \frac{\alpha(t - t_0)}{A(t)R} \int_0^{2\pi} r^4(\xi, t_0) \mathbf{n}(\xi) d\xi + \frac{1}{3A(t)} \int_0^{2\pi} \left(\frac{\partial^2 r^3(\xi, t_0)}{2\partial t^2}(t - t_0)^2 + \dots \right) \mathbf{n}(\xi) d\xi, \\
\overset{(II)}{=} \mathbf{c}(t_0) &- \frac{\alpha(t - t_0)}{A(t)R} \int_0^{2\pi} r^3(\xi, t_0) \Bigg(r(\xi_0, t_0) + \frac{\partial r(\xi_0, t_0)}{\partial \xi}(\xi - \xi_0) \\
&+ \frac{\partial^2 r(\xi_0, t_0)}{2\partial \xi^2}(\xi - \xi_0)^2 + \dots \Bigg) \mathbf{n}(\xi) d\xi \\
&\quad + \frac{1}{3A(t)} \int_0^{2\pi} \left(\frac{\partial^2 r^3(\xi, t_0)}{2\partial t^2}(t - t_0)^2 + \dots \right) \mathbf{n}(\xi) d\xi, \\
= \mathbf{c}(t_0) &- \frac{\alpha(t - t_0)}{A(t)R} \frac{\partial r(\xi_0, t_0)}{\partial \xi} \int_0^{2\pi} r^3(\xi, t_0) \xi \mathbf{n}(\xi) d\xi \\
&+ \frac{\alpha(t - t_0)}{A(t)R} \int_0^{2\pi} \left(\frac{\partial^2 r(\xi_0, t_0)}{2\partial \xi^2}(\xi - \xi_0)^2 + \dots \right) r^3(\xi, t_0) \mathbf{n}(\xi) d\xi \\
&\quad + \frac{1}{3A(t)} \int_0^{2\pi} \left(\frac{\partial^2 r^3(\xi, t_0)}{2\partial t^2}(t - t_0)^2 + \dots \right) \mathbf{n}(\xi) d\xi, \\
\overset{(III)}{=} \mathbf{c}(t_0) &- \frac{\alpha(t - t_0)}{A(t)R} \frac{\partial r(\xi_0, t_0)}{\partial \xi} \left(\left[\xi \int_0^{\xi} r^3(\xi', t_0) \mathbf{n}(\xi') d\xi' \right]_0^{2\pi} - \int_0^{2\pi} r^3(\xi, t_0) \mathbf{n}(\xi) d\xi \right) \\
&+ \frac{\alpha(t - t_0)}{A(t)R} \int_0^{2\pi} \left(\frac{\partial^2 r(\xi_0, t_0)}{2\partial \xi^2}(\xi - \xi_0)^2 + \dots \right) r^3(\xi, t_0) \mathbf{n}(\xi) d\xi \\
&\quad + \frac{1}{3A(t)} \int_0^{2\pi} \left(\frac{\partial^2 r^3(\xi, t_0)}{2\partial t^2}(t - t_0)^2 + \dots \right) \mathbf{n}(\xi) d\xi, \\
= \mathbf{c}(t_0) &+ \frac{\alpha(t - t_0)}{A(t)R} \int_0^{2\pi} \left(\frac{\partial^2 r(\xi_0, t_0)}{2\partial \xi^2}(\xi - \xi_0)^2 + \dots \right) r^3(\xi, t_0) \mathbf{n}(\xi) d\xi \\
&+ \frac{1}{3A(t)} \int_0^{2\pi} \left(\frac{\partial^2 r^3(\xi, t_0)}{2\partial t^2}(t - t_0)^2 + \dots \right) \mathbf{n}(\xi) d\xi,
\end{aligned} \tag{31}$$

where (I) holds by substitution of (16), (II) holds by substitution of a Taylor series of $r(\xi, t_0)$ around some ξ_0, and (III) holds by using integration by parts as $\int u dv = uv - \int v du$ with $u = \xi$ and $dv = r^3(\xi, t_0)\mathbf{n}(\xi)$. Additionally, at various points the previously found property $\int_0^{2\pi} r^3(\xi, t_0)\mathbf{n}(\xi) d\xi = \mathbf{0}$ is used, which follows from (26). In general, $r(\xi, t_0)$ may not be perfectly circular, however, it is likely that parts of the contour are circular. Consequently, r is constant over a small segment of ξ, and ξ_0 may now be chosen such that higher order derivatives in ξ_0 with respect to ξ equal zero. Thus, as a result, (31) reduces to:

$$\mathbf{c}(t) = \mathbf{c}(t_0) + \frac{(t - t_0)^2}{6A(t)} \int_0^{2\pi} \left(\frac{\partial^2 r^3(\xi, t_0)}{\partial t^2} + \dots \right) \mathbf{n}(\xi) d\xi. \tag{32}$$

Note that, in comparison with the first substitution of the Taylor series around t_0, the first two terms have dropped. It is clear now that the shift in $\mathbf{c}(t)$ is dependent mostly on the second derivative of $r^3(\xi, t_0)$ with respect to t. Additionally, it can be shown through repeated use of the chain rule and (16) that:

$$\frac{\partial^2 r^3(\xi, t_0)}{\partial t^2} = \alpha^2 \left(9r^3(\xi, t_0) - 21\frac{r^4(\xi, t_0)}{R} + 12\frac{r^5(\xi, t_0)}{R^2} \right). \tag{33}$$

The $r^3(\xi, t_0)$ terms cancel again, based on the observation in (26), such that:

$$\mathbf{c}(t) = \mathbf{c}(t_0) + \frac{(t-t_0)^2}{6A(t)} \int_0^{2\pi} \left(\frac{\alpha^2}{R} \left(12R^{-1}r^5(\xi, t_0) - 21r^4(\xi, t_0) \right) + \dots \right) \mathbf{n}(\xi) d\xi. \tag{34}$$

Since α denotes growth of r per units t, $\alpha(t - t_0)$ scales with how much the radius can grow between two instances of the time-series, which should always be small for a sensible time-series. Additionally, the area of the crop $A(t)$ and the maximum achievable radius R are magnitudes larger. As a result, $\alpha^2(t-t_0)^2(A(t)R)^{-1}$ must be small.

3.2.2 *Keypoint Descriptor*

Having selected the centroid as the keypoint's position $\mathbf{p}_i$, see (23), this section will introduce two different possible descriptors $\mathbf{f}_i$ that may be used to describe each keypoint, or crop, i. The first descriptor is based on the shape and size of the crop in question, while the second one considers the local pattern in which crops are planted.

Shape and Size Based Descriptor

First, a naive descriptor vector is setup using simply the radius of crop i at n different angles ξ_j:

$$\mathbf{f}_i = \left\{ r_i(\xi_j) \;\middle|\; \xi_j = j\frac{2\pi}{n} \text{ for } j = 1, ..., n \right\}, \tag{35}$$

An example of such a naive descriptor based on the contour in figure 5 is given in figure 6.

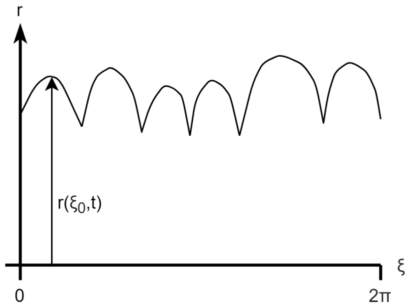


Figure 6: Naive shape and size based descriptor as defined in (35) for the contour given in figure 5

Note that this descriptor fails to be time-invariant, as the radius is expected to increase over time by (16). Based on the observation in (22), a global normalisation per orthophoto will be introduced. For *one* orthophoto, let the set F of all naive descriptor values be given as:

$$F = \left\{ f_{i,j} \mid \forall i, j \right\}. \tag{36}$$

Next, a lookup table is created as the vector $\mathbf{l}$:

$$\mathbf{l} = \{F_\nu \mid \nu = 1, ..., 100\}, \tag{37}$$

where F_ν denotes the ν-th percentile of the set F. Finally, a normalisation can be applied to each descriptor in the orthophoto by revising each element as:

$$f_{i,j} \leftarrow \nu \qquad \text{if } l_{\nu-1} < f_{i,j} \leqslant l_\nu, \tag{38}$$

where it may be used that $l_0 = 0$. An example of this revision step for the naive descriptor presented in figure 6 is given in figure 7.

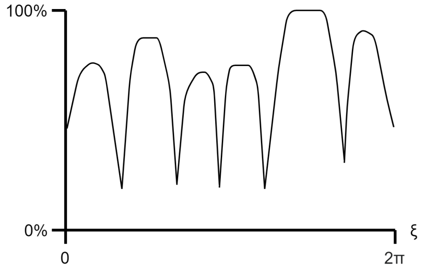


Figure 7: Result of revising the naive descriptor given in figure 6 by (38) using some l based on the set of all contours present in the orthophoto, see (36) and (37).

Note that in the example in figure 7 there is a domain of ξ for which the revised descriptor equals 100. This implies that the radius found at this domain of ξ is strictly larger than 99 % of other radii found in the orthophoto thus being in the 100th quantile. Based on the observation in (22) it is expected that for any given moment in the future, the radii for this contour in this domain of ξ remain in the 100th quantile.

Planting Pattern Based Descriptor

Alternatively, a descriptor can be constructed based solely on the pattern in which crops are planted. Consider orthophoto A with crops $i \in S^A$, a descriptor vector of dimension n for crop or keypoint i, is created by:

$$\mathbf{f}_i = \left\{ \left\| \mathbf{p}_i - \mathbf{p}_{k_j} \right\|_2 \;\middle|\; k_j = \operatorname*{arg\,min}_{k \in S^A \setminus \{k_1, \dots, k_{j-1}\}} \|\mathbf{p}_i - \mathbf{p}_k\|_2 \,,\ \text{for } j = 1, \dots, n \right\}. \tag{39}$$

That is, a descriptor $\mathbf{f}_i$ is created by storing the sorted distances to the n nearest identified crops. An schematic example using $n = 5$ is given in figure 8.

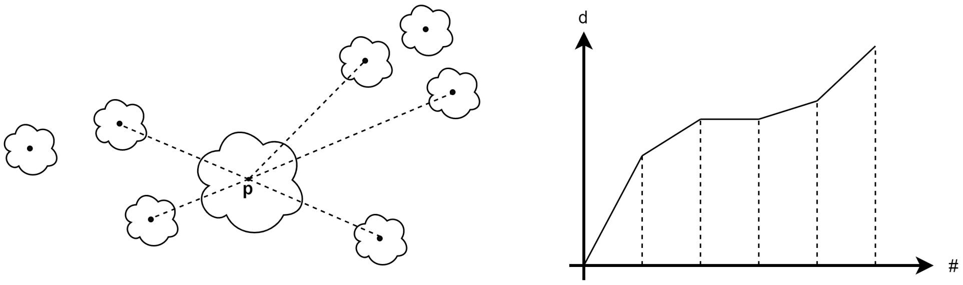


Figure 8: Example of the pattern based descriptor for some keypoint, or contour, with position $\mathbf{p}$ using $n = 5$. The distances to the n nearest other contours are gathered and sorted creating the descriptor according to (39) given on the right-hand side.

Under the error present between orthophoto A and B, see (14), these distances should be preserved between two orthophotos A and B, thus making (39) a suitable descriptor. To see why this is true, consider a set of n corresponding points $(i \leftrightarrow \iota)$

with $\mathbf{x}_i$ a point in orthophoto A corresponding to $\mathbf{x}_\iota$ in orthophoto B. By (14) it is true that for each $(i \leftrightarrow \iota)$:

$$\mathbf{x}_i^A = \mathbf{t} + R\mathbf{x}_\iota^B + \boldsymbol{w}(\mathbf{x}_\iota^B). \tag{40}$$

Let the translation, rotation and scaling be defined through:

$$(\mathbf{t}, R) = \underset{(\tilde{\mathbf{t}},\tilde{R})}{\arg\min} \sum_{(i \leftrightarrow \iota)} \left\| \mathbf{x}_i^A - \tilde{\mathbf{t}} - \tilde{R}\mathbf{x}_\iota^B \right\|_2^2. \tag{41}$$

Writing $\mathbf{t} = \langle t_1, t_2 \rangle$ and $R = [r_1, -r_2; r_2, r_1]$, the minimisation problem in (41) can be formulated as an over-determined linear system $A\mathbf{x} = \mathbf{b}$ as:

$$\begin{bmatrix} I & P_1 \\ I & P_2 \\ \vdots & \vdots \\ I & P_n \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^A \\ \mathbf{x}_2^A \\ \vdots \\ \mathbf{x}_n^A \end{bmatrix}, \tag{42}$$

with:

$$P_\iota = \begin{bmatrix} x_\iota^B & -y_\iota^B \\ y_\iota^B & x_\iota^B \end{bmatrix}. \tag{43}$$

The least-squares solution $\hat{\mathbf{x}}$ to the system $A\mathbf{x} = \mathbf{b}$ has as property that $A^T(\mathbf{b} - A\hat{\mathbf{x}}) = \mathbf{0}$. Since $\mathbf{t}$ and R are chosen as such a least-squares solution to the system given in (42), it follows -in combination with (40)- that:

$$\begin{bmatrix} I & P_1 \\ I & P_2 \\ \vdots & \vdots \\ I & P_n \end{bmatrix}^T \begin{bmatrix} \boldsymbol{w}(\mathbf{x}_1^B) \\ \boldsymbol{w}(\mathbf{x}_2^B) \\ \vdots \\ \boldsymbol{w}(\mathbf{x}_n^B) \end{bmatrix} = \mathbf{0}, \tag{44}$$

$$\begin{bmatrix} 1 & 0 & 1 & 0 & \dots & 1 & 0 \\ 0 & 1 & 0 & 1 & \dots & 0 & 1 \\ x_1^B & y_1^B & x_2^B & y_2^B & \dots & x_n^B & y_n^B \\ -y_1^B & x_1^B & -y_2^B & x_2^B & \dots & -y_n^B & x_n^B \end{bmatrix} \begin{bmatrix} u(\mathbf{x}_1^B) \\ v(\mathbf{x}_1^B) \\ \vdots \\ u(\mathbf{x}_n^B) \\ v(\mathbf{x}_n^B) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{45}$$

where $\boldsymbol{w}(\mathbf{x}) = \langle u(\mathbf{x}), v(\mathbf{x}) \rangle$. Based on the first and second row combined, third row, and fourth row in (45), one has respectively:

$$\sum_{\iota=1}^{n} \boldsymbol{w}(\mathbf{x}_\iota^B) = \mathbf{0}, \tag{46}$$

$$\sum_{\iota=1}^{n} \left(\mathbf{x}_\iota^B\right)^T \boldsymbol{w}(\mathbf{x}_\iota^B) = 0, \tag{47}$$

$$\sum_{\iota=1}^{n} \left(R_o \mathbf{x}_\iota^B\right)^T \boldsymbol{w}(\mathbf{x}_\iota^B) = 0, \tag{48}$$

where $R_o \langle x, y \rangle = \langle -y, x \rangle$, therefore $R_0 = [0, -1; 1, 0]$. Based on these properties:

$$\begin{aligned} \sum_{i=1}^{n} \sum_{\tilde{\imath}=1}^{n} \left\| \mathbf{x}_i^A - \mathbf{x}_{\tilde{\imath}}^A \right\|_2^2 &= \sum_{\iota=1}^{n} \sum_{\tilde{\imath}=1}^{n} \left\| R\left(\mathbf{x}_\iota^B - \mathbf{x}_{\tilde{\imath}}^B\right) + \boldsymbol{w}(\mathbf{x}_\iota^B) - \boldsymbol{w}(\mathbf{x}_{\tilde{\imath}}^B) \right\|_2^2, \\ &= \sum_{\iota=1}^{n} \sum_{\tilde{\imath}=1}^{n} \Bigg(\left(\mathbf{x}_\iota^B - \mathbf{x}_{\tilde{\imath}}^B\right)^T R^T R \left(\mathbf{x}_\iota^B - \mathbf{x}_{\tilde{\imath}}^B\right) \\ &\quad + \left\| \boldsymbol{w}(\mathbf{x}_\iota^B) - \boldsymbol{w}(\mathbf{x}_{\tilde{\imath}}^B) \right\|_2^2 + 2\left(\boldsymbol{w}(\mathbf{x}_\iota^B) - \boldsymbol{w}(\mathbf{x}_{\tilde{\imath}}^B)\right)^T R \left(\mathbf{x}_\iota^B - \mathbf{x}_{\tilde{\imath}}^B\right) \Bigg), \\ &= \left(r_1^2 + r_2^2\right) \sum_{\iota=1}^{n} \sum_{\tilde{\imath}=1}^{n} \left\| \mathbf{x}_\iota^B - \mathbf{x}_{\tilde{\imath}}^B \right\|_2^2 + \sum_{\iota=1}^{n} \sum_{\tilde{\imath}=1}^{n} \left\| \boldsymbol{w}(\mathbf{x}_\iota^B) - \boldsymbol{w}(\mathbf{x}_{\tilde{\imath}}^B) \right\|_2^2 \end{aligned} \tag{49}$$

where in the last line it was used that $R^T R = (r_1^2 + r_2^2)I$ and $R = r_1 I + r_2 R_o$, which, in the latter case, causes the last term to cancel by (47) and (48). Notice that the $(r_1^2 + r_2^2)$ term corresponds to scaling. For pure rotation by $r_1 = \cos\theta$ and $r_2 = \sin\theta$, this term equals one. However, since r_1 and r_2 are selected as the least-squares solution to (42), it is possible to have some scaling depending on the points correspondences that are used. The distortion in the mutual distances between points is given by:

$$\begin{aligned}\sum_{\iota=1}^{n}\sum_{\tilde{\iota}=1}^{n}\left\|\boldsymbol{w}(\mathbf{x}_{\iota}^{B})-\boldsymbol{w}(\mathbf{x}_{\tilde{\iota}}^{B})\right\|_2^2 &= \sum_{\iota=1}^{n}\sum_{\tilde{\iota}=1}^{n}\left\|\boldsymbol{w}(\mathbf{x}_{\iota}^{B})\right\|_2^2 + \sum_{\iota=1}^{n}\sum_{\tilde{\iota}=1}^{n}\left\|\boldsymbol{w}(\mathbf{x}_{\tilde{\iota}}^{B})\right\|_2^2 \\ &\quad - 2\sum_{\iota=1}^{n}\underbrace{\sum_{\tilde{\iota}=1}^{n}\boldsymbol{w}(\mathbf{x}_{\tilde{\iota}}^{B})^T}_{0^T \text{ by } (46)}\boldsymbol{w}(\mathbf{x}_{\iota}^{B}), \\ &= 2n\sum_{\iota=1}^{n}\left\|\boldsymbol{w}(\mathbf{x}_{\iota}^{B})\right\|_2^2 .\end{aligned} \tag{50}$$

Notice that this distortion is related to the 'standard deviation' in the sampled points of the distortion field $\boldsymbol{w}$ around its mean of zero, see (46). Indeed, dividing by the total number of mutual distances, n^2, one finds:

$$\text{mean}\left(\left\|\mathbf{x}_i^A - \mathbf{x}_{\tilde{i}}^A\right\|_2^2\right) = (r_1^2 + r_2^2)\text{mean}\left(\left\|\mathbf{x}_{\iota}^B - \mathbf{x}_{\tilde{\iota}}^B\right\|_2^2\right) + 2\sigma_{\boldsymbol{w}}^2, \tag{51}$$

where the variance $\sigma_{\boldsymbol{w}}^2$ of the residual distortion vectors $\boldsymbol{w}(\mathbf{x}_{\iota}^B)$ is defined as:

$$\sigma_{\boldsymbol{w}}^2 = \frac{1}{n}\sum_{\iota=1}^{n}\left\|\boldsymbol{w}(\mathbf{x}_{\iota}^{B})\right\|_2^2, \tag{52}$$

and the 'mean' operator denotes the arithmetic mean. Thus, after applying a global correction of the error – using $\mathbf{t}$ and R – the mean mutual squared distance between points in orthophoto A will remain larger than its counterpart in the corrected orthophoto B_c by the amount that equals two times the variance of the residual vectors $\boldsymbol{w}(\mathbf{x}_i^B)$. For the given dataset, $\boldsymbol{w}$ was found to be of a small magnitude, hence, the amount that $\mathbf{f}_i$, as defined in (39), differs for different orthophotos is expected to be small.

3.2.3 *Keypoint Matching*

Keypoint matching may be done in a comparable way to what was presented in literature, however, with the added constraint of locality. The geotransform provided with the orthophotos is incorrect, but only up to a certain extent, which provides an upper bound in possible spatial distance between keypoints.

Consider orthophoto A with keypoints $i \in S^A$, and orthophoto B with keypoints that have index $i \in S^B$. Each keypoint again has a location $\mathbf{p}_i$ and descriptor $\mathbf{f}_i$ assigned to it, provided now by the methods previously proposed. For some keypoint $i \in S^A$, a set of nearby keypoints τ in orthophoto B can be defined as:

$$S^B(i, D) = \left\{\tau \;\middle|\; \tau \in S^B, \|\mathbf{p}_i - \mathbf{p}_\tau\|_2 < D\right\}. \tag{53}$$

Here, D provides an upper bound of the shift in geotransform between two orthophotos. In practice, D is dependent on the accuracy of the GPS built into the camera. Given this group $S^B(i, D)$ of nearby keypoints, matching may proceed as described in literature by defining a candidate match as:

$$\iota = \underset{\tau \in S^B(i,D)}{\arg\min}\ \|\mathbf{f}_i - \mathbf{f}_\tau\|_1, \tag{54}$$

and accepting this candidate as match, if and only if:

$$\|\mathbf{f}_i - \mathbf{f}_\iota\|_1 < C_1 \min_{\tau \in S^B(i,D)\setminus\{\iota\}} \|\mathbf{f}_i - \mathbf{f}_\tau\|_1 \,, \tag{55}$$

where $C_1 \leqslant 1$ is a constant. Note the slight differences between (54) and (11), and (55) and (12).
Equation (55) allows one to check that ι is a 'unique' match to i, however, it -presumably- does not check whether i is an unique match to ι. Suppose that we have $i, j \in S^A$ that are similar keypoints. That is, $\mathbf{f}_i \approx \mathbf{f}_j$. By the proposed method for matching, both i and j may be matched to the same $\iota \in S^B$. To avoid this, it may prove useful to apply the same criterion backwards, and accept the match only if -in addition to (55)- also:

$$\|\mathbf{f}_\iota - \mathbf{f}_i\|_1 < C_2 \min_{\tau \in S^A(i,D)\setminus\{i\}} \|\mathbf{f}_\iota - \mathbf{f}_\tau\|_1 \,, \tag{56}$$

Again, $C_2 \leqslant 1$ is a constant. Setting $C_2 = 1$ is equivalent to not using this additional check.

3.2.4 *Model Fitting*

Having a set of matched keypoints between the two orthophotos A and B, the error characterised in $\mathbf{t}$, R, and $\boldsymbol{w}$ may be derived such that it can be corrected for. For a number of iterations, a random sample is drawn from the set of all matches. The required sample size, used only to estimate $\mathbf{t}$ and R, is three. Corresponding to the drawn random sample, a candidate $\tilde{\mathbf{t}}$ and $\tilde{R}$ are found, which are tested against all data by checking how many matches are 'inliers', i.e., satisfy:

$$\left\|\mathbf{p}_i - \left(\tilde{\mathbf{t}} + \tilde{R}\mathbf{p}_\iota\right)\right\|_2 < \varepsilon, \tag{57}$$

for some threshold error ε. The candidates set $\tilde{\mathbf{t}}$ and $\tilde{R}$ the largest number of inliers after all iterations is saved as the true $\mathbf{t}$ and R. Alternatively, $\mathbf{t}$ and R may be selected as the least-squares fit on the largest set of inliers. Next, the residual vector field $\boldsymbol{w}$ is determined by collecting all $\boldsymbol{w}(\mathbf{x}_\iota^B)$ and applying for example a piece-wise linear, or a polynomial fit to it. Notice that a piece-wise linear fit is exact in correctly matches points, however, it may not be smooth, thus a second degree polynomial fit for $\boldsymbol{w}$ was used in the current implementation. Finally, orthophoto B may be corrected by $\mathbf{x}^{B_c} = \mathbf{t} + R\mathbf{x}^B + \boldsymbol{w}(\mathbf{x}^B)$.

4 TIME-SERIES DATASET

The dataset used for development and testing of the proposed methods is a time-series of a potato field. The field, that is, the area in which crops are planted, is approximately 75 by 250 meters. The time-series contains a total of seven instances, or orthophotos, divided between May and June of 2019, each created using around 300 images. These images were captured using a *DJI Phantom 4 Pro* [3] at an altitude of 30 meters, and processed into orthophotos using the open-source photogrammetry pipeline *WebODM* by *OpenDroneMap* [4]. The resulting orthophotos have a ground sampling distance of 1 cm, that is, one pixel contains a square 1 cm^2 area.
A single image corresponding to one of the orthophotos is shown in figure 9. The settings used for creation of the orthophotos with WebODM are given in table 1. Finally, an overview of the resulting orthophotos in the time-series, as well as a representation of the growth stages that were captured, is given in figure 10.

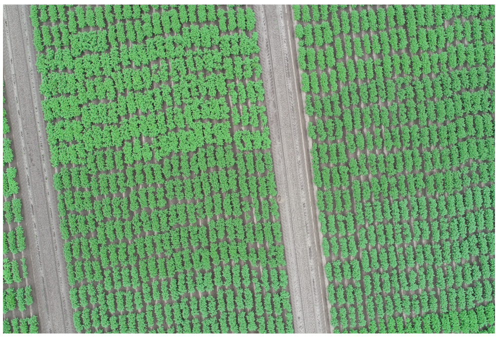


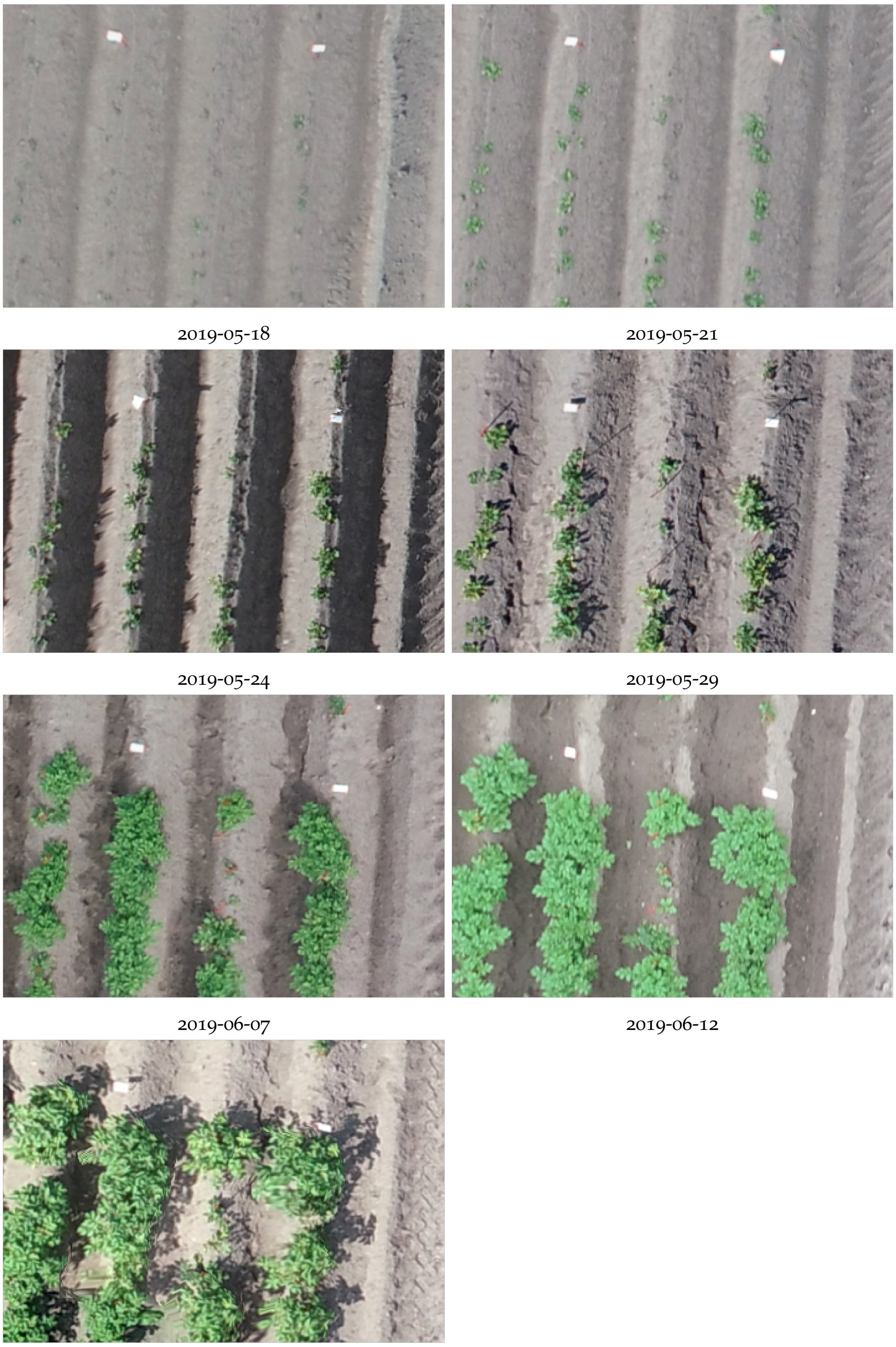
Figure 9: A 5472 × 3648 image corresponding to the 2019-06-12 orthophoto in the time-series. Captured by the DJI Phantom 4 Pro at an altitude of 30 meters.

mesh-octree-depth	12
min-num-features	10000
resize-to	-1
texturing-nadir-weight	0
orthophoto-resolution	1
dem-resolution	1
ignore-gsd	true
build-overviews	true
crop	0
camera-lens	brown
skip-3dmodel	true
depthmap-resolution	1280

Table 1: Parameter settings for WebODM used to create orthophotos in the time-series. The dataset for each instance consists of roughly 300 separate images with a certain overlap.

[3] https://www.dji.com/phantom-4-pro/info

[4] https://github.com/OpenDroneMap/WebODM



2019-05-18

2019-05-21

2019-05-24

2019-05-29

2019-06-07

2019-06-12

2019-06-19

Figure 10: Orthophotos created by WebODM using the settings given in table 1, and the images captured by the DJI Phantom 4 Pro. The varying growth stages in this time-series are visualised using a fixed zoom-level showing *roughly* the same location in each instance.

5 RESULTS

Seen as how descriptor matching is crucial to the proposed method, this part will be tested separately first, after which the full method is tested. In order to enable both these testing purposes, a ground-truth is created first.

5.1 Ground-Truth

Ground-truth is established by leveraging a total of approximately 1300 small white cards, coincidentally present in the field. Using their location, the error $\mathbf{t}, R, \boldsymbol{w}$ may be found. Consider some orthophoto A. The orthophoto (R^A, G^A, B^A) is translated to a binary map I^A by:

$$I^A(\mathbf{x}) = \begin{cases} 1, & \text{if } R^A(\mathbf{x}) > t_r,\ G^A(\mathbf{x}) > t_g,\ B^A(\mathbf{x}) > t_b, \\ 0, & \text{otherwise.} \end{cases} \tag{58}$$

Here, t_r denotes a threshold value for the red channel, etc. In the current implementation for *white* cards, $t_r = t_g = t_b = 235$ was chosen.

One may think of I^A as a large connected graph, in which each pixel corresponds to a node, that is connected eight -or four- of its neighbouring pixels by an edge. Deleting all nodes for which $I^A(\mathbf{x}) = 0$, the remaining connected components corresponding to $I^A(\mathbf{x}) = 1$ can be extracted, which should correspond to the white cards.

Each component i is a set of pixels again, denoted by C_i. For orthophoto A, let $i \in S^A$. The position $\mathbf{p}_i$ is selected as the centroid of the set C_i, defined as the arithmetic mean of all pixels in C_i. First, however, all components are filtered by area:

$$\textbf{if } |C_i| < A_{\text{min}} \text{ or } |C_i| > A_{\text{max}} \textbf{ then} \text{ discard } C_i. \tag{59}$$

In the case of the white cards, $A_{\text{min}} = 10$ cm² and $A_{\text{max}} = 500$ cm², corresponding to the expected size of the card. Now, consider also orthophoto B. Similar to before, I^B is constructed and a number of white cards i with $i \in S^B$ are extracted. For $i \in S^A$, a set of matches $\iota \in S^B$ is defined by:

$$\iota = \left\{ \tau \,\middle|\, \|\mathbf{p}_i - \mathbf{p}_\tau\|_1 < D, \tau \in S^B \right\}. \tag{60}$$

Here, D corresponds to the maximum shift between two orthophotos one expects to find. In current implementation it was set empirically as $D = 500$, i.e. 5 meters. Note that, contrary to how ι was defined in (11) or (54), ι may now have more than one member. Generally speaking, however, this is not the case in the present scenario as the spacing between *different* cards in the same time instance exceeds the value of D. Consequently, in general, a card is matched to precisely one card in orthophoto B.

The resulting matches can be processed by RANSAC, as presented previously in (57), such that orthophoto B may be corrected by $\mathbf{x}^{B_c} = \mathbf{t} + R\mathbf{x}^B + \boldsymbol{w}(\mathbf{x}^B)$.

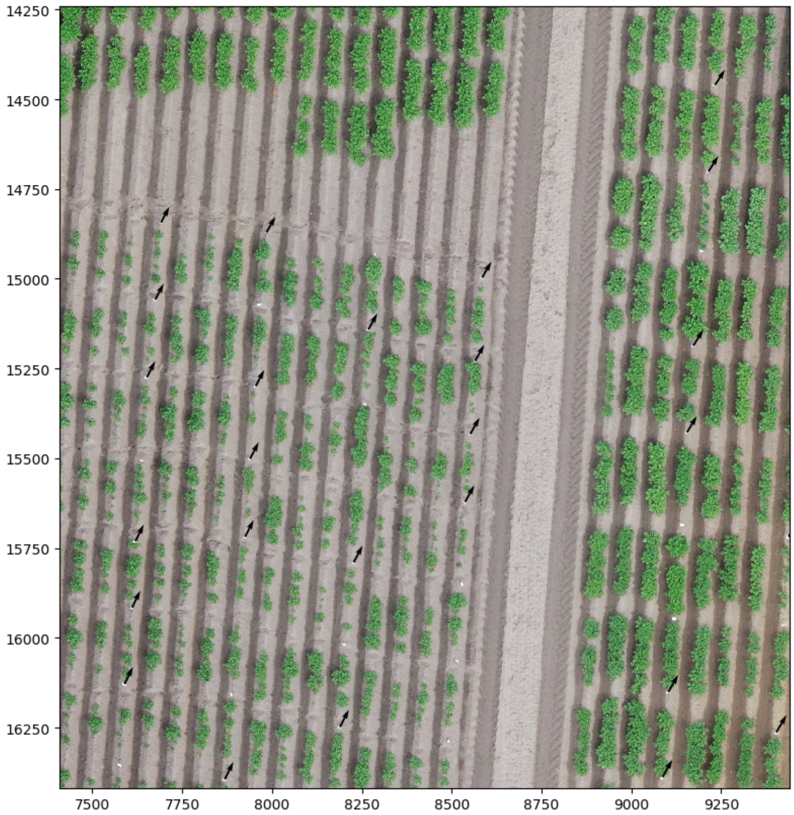


Figure 11: Displacement vector, or error, between white cards on orthophotos created on two different dates. On a whole the error may be characterised by translation, a small rotation, and possible minor local deformations.

In figure 11, the displacement vector between matched white cards is shown. In general it was noted that the error mostly consisted of translation and some rotation, which is clearly visible in the figure, although some minor local deformations were found as well.

5.2 Descriptor Matching

For testing of the *matching* procedure, the ground-truth is used beforehand, meaning that matching is tested between two orthophotos that are *not* misaligned. As a consequence, a correct match i with ι should always yield $\|\mathbf{p}_i - \mathbf{p}_\iota\|_2$ equal to zero. This property can be used to check the number of correct matches before using RANSAC model fitting.
Let N_T be the total number of matches generated by the method that satisfy (55). Within those N_T correspondences, let N_C be the number of matches that is correct, using that $\|\mathbf{p}_i - \mathbf{p}_\iota\|_2$ should equal zero. Next, let N_R be the number of matches that RANSAC designates to be an inlier according to (57) for the best $\tilde{\mathbf{t}}$ and $\tilde{R}$. Finally, let N_{RC} be the number of matches that is considered both an inlier by RANSAC and also correct by ground-truth.
For both introduced descriptors, see (35) and (38) as well as (39), the effectiveness of the method under different parameter settings is tested by changing one parameter at a time.

Shape and Size Based Descriptor

Consider the shape and size based descriptor introduced in (35) and (38). Let A_{min} and A_{max} refer to preprocessing thresholds used to discard certain identified potential crops. Let the newly introduced parameter n_f be the sampling factor, used as $n = \lfloor n_f L \rfloor$ with n as defined in (35) and L being the average length of a crop contour in orthophoto A. Furthermore, w refers to the number of increments between zero and 100 % in (37). Finally, D, C_1, C_2, and ε are defined in (53), (55), (56), and (57), respectively. The default parameter settings for the shape and size based descriptor are given in table 2.

A	2019-05-21	
B	2019-05-24	
A_{min}	100	cm^2
A_{max}	1000	cm^2
n_f	2	
w	100	
D	500	cm
C_1	0.8	
C_2	1.0	
ε	10	cm

Table 2: Default parameter settings for the proposed method using the shape and size based descriptor.

Next, one parameter is varied at a time, keeping other parameters constant. Results of testing are summarised in tables 3 through 12.

A	05-18	*05-21*	05-24	05-29	06-07	06-12
B	05-21	*05-24*	05-29	06-07	06-12	06-19
N_T	116	*789*	782	4712	1157	595
N_C	16	*366*	92	36	105	66
N_R	16	*361*	76	86	14	14
N_{RC}	16	*347*	74	0	11	11

Table 3: Effectiveness of the shape and size based descriptor while varying date, taking a step of size one in the time-series. All other parameters are kept as indicated in table 2, with the exception of using $A_{\mathrm{min}} = 15$ in the first column, and $A_{\mathrm{max}} = 5000$ in the last three columns.

A	05-18	05-21	05-24	05-29	06-07
B	05-24	05-29	06-07	06-12	06-19
N_T	20	511	2045	6285	1517
N_C	6	54	7	11	12
N_R	6	7	149	130	23
N_{RC}	5	4	0	0	1

Table 4: Effectiveness of the shape and size based descriptor while varying date, taking a step of size two in the time-series. All other parameters are kept as indicated in table 2, with the exception of using $A_{\mathrm{max}} = 5000$ in the last three columns.

A_{min}	0	5	15	50	*100*	150	200	250	500
N_T	-	-	2247	1474	*789*	425	249	137	7
N_C	-	-	533	498	*366*	230	171	102	5
N_R	-	-	560	512	*361*	201	165	103	6
N_{RC}	-	-	532	486	*347*	201	159	102	5

Table 5: Effectiveness of the shape and size based descriptor while varying A_{min}. All other parameters are kept as indicated in table 2. Note the failure of the method for $A_{\text{min}} \leqslant 5$ as unsuitable shapes are included such as lines.

A_{max}	150	300	500	750	*1000*	5000	10^4	10^5	∞
N_T	399	702	759	779	*789*	788	788	788	788
N_C	64	232	339	363	*366*	367	367	367	367
N_R	70	237	350	377	*361*	371	371	371	371
N_{RC}	64	225	332	359	*347*	362	362	362	362

Table 6: Effectiveness of the shape and size based descriptor while varying A_{max}. All other parameters are kept as indicated in table 2. Note that increasing beyond a certain point may not change the results, while it does increase computation time.

n_f	0.1	0.5	1.0	1.5	*2.0*	2.5	3.0	4.0	5.0
N_T	1585	845	789	794	*789*	794	795	795	792
N_C	197	354	357	356	*366*	359	360	362	357
N_R	76	369	374	373	*361*	377	375	379	374
N_{RC}	69	354	355	355	*347*	358	357	361	356

Table 7: Effectiveness of the shape and size based descriptor while varying n_f. All other parameters are kept as indicated in table 2. The setting $n_f = 2$, along with the parameters provided in table 2, leads to $n = 97$.

w	2	5	25	50	*100*	150	255	500	1000
N_T	1365	902	790	793	*789*	789	782	782	780
N_C	299	343	359	365	*366*	364	364	363	363
N_R	264	358	376	382	*361*	381	374	380	377
N_{RC}	263	343	357	362	*347*	363	356	362	359

Table 8: Effectiveness of the shape and size based descriptor while varying w. All other parameters are kept as indicated in table 2.

D	100	200	300	400	*500*	800	1100	1400	1700
N_T	1897	1343	1081	861	*789*	608	517	446	415
N_C	1309	812	581	449	*366*	249	178	133	110
N_R	1378	852	564	470	*361*	250	185	135	113
N_{RC}	1306	811	561	448	*347*	240	178	128	107

Table 9: Effectiveness of the shape and size based descriptor while varying D. All other parameters are kept as indicated in table 2. Note that low values of D are possible as ground-truth was used beforehand.

C_1	0.4	0.5	0.6	0.7	*0.8*	0.85	0.9	0.95	1.0
N_T	33	80	174	370	*789*	1146	1618	2273	3220
N_C	29	64	122	216	*366*	452	540	637	747
N_R	29	66	126	225	*361*	476	560	626	746
N_{RC}	29	64	122	216	*347*	451	536	595	738

Table 10: Effectiveness of the shape and size based descriptor while varying C_1. All other parameters are kept as indicated in table 2.

C_1	*0.8*	0.85	0.9	0.95	1.0
C_2	*1.0*	0.95	0.9	0.85	0.8
N_T	*789*	866	864	800	723
N_C	*366*	352	308	263	201
N_R	*361*	368	316	274	211
N_{RC}	*347*	351	304	263	201

Table 11: Effectiveness of the shape and size based descriptor while varying C_1 and C_2 simultaneously. All other parameters are kept as indicated in table 2.

ε	1	2	5	8	*10*	50	100	200	500
N_T	789	789	789	789	*789*	789	789	789	789
N_C	366	366	366	366	*366*	366	366	366	366
N_R	104	242	345	364	*361*	394	423	483	784
N_{RC}	102	242	344	348	*347*	366	366	366	366

Table 12: Effectiveness of the shape and size based descriptor while varying ε. All other parameters are kept as indicated in table 2.

Planting Pattern Based Descriptor

Consider the planting pattern based descriptor introduced in (39). Let A_{min} and A_{max} refer -as before- to preprocessing thresholds used to discard certain identified potential crops. Furthermore, let n, D, C_1, C_2, and ε be as defined in (39), (53), (55), (56), and (57), respectively. The default parameter settings for the planting pattern based descriptor are given in table 13.

A	2019-05-21	
B	2019-05-24	
A_{min}	5	cm^2
A_{max}	10^4	cm^2
n	4	
D	500	cm
C_1	0.8	
C_2	1.0	
ε	10	cm

Table 13: Default parameter settings for the proposed method using the planting pattern based descriptor.

Next, one parameter is varied at a time, keeping other parameters constant. Results of testing are summarised in tables 14 through 22.

A	05-18	05-21	05-24	05-29	06-07	06-12
B	05-21	05-24	05-29	06-07	06-12	06-19
N_T	390	8412	9727	9916	4199	4355
N_C	40	3538	961	104	178	232
N_R	40	3721	160	20	21	41
N_{RC}	40	3532	157	0	17	37

Table 14: Effectiveness of the planting pattern based descriptor while varying date, taking a step of size one in the time-series. All other parameters are kept as indicated in table 13.

A	05-18	05-21	05-24	05-29	06-07
B	05-24	05-29	06-07	06-12	06-19
N_T	353	6773	9025	10187	3961
N_C	26	530	63	25	36
N_R	12	68	45	81	13
N_{RC}	10	64	0	0	0

Table 15: Effectiveness of the planting pattern based descriptor while varying date, taking a step of size two in the time-series. All other parameters are kept as indicated in table 13.

A_{min}	1	2	3	4	*5*	7	10	20	100
N_T	8250	8607	8560	8565	*8412*	8258	7943	6911	1694
N_C	1045	3151	3103	3619	*3538*	3538	3409	2782	439
N_R	96	3310	3238	3759	*3721*	3424	3567	2932	410
N_{RC}	90	3143	3073	3565	*3532*	3254	3404	2780	410

Table 16: Effectiveness of the planting pattern based descriptor while varying A_{min}. All other parameters are kept as indicated in table 13.

A_{max}	100	500	1000	5000	10^4	10^5	10^6	10^7	∞
N_T	5878	8387	8408	8412	*8412*	8412	8412	8412	8412
N_C	828	3482	3536	3538	*3538*	3538	3538	3538	3538
N_R	670	3650	3719	3721	*3721*	3721	3721	3721	3721
N_{RC}	666	3465	3530	3532	*3532*	3532	3532	3532	3532

Table 17: Effectiveness of the planting pattern based descriptor while varying A_{max}. All other parameters are kept as indicated in table 13. Note that on later dates, the setting $A_{max} = 10^4$ did prove useful.

n	1	2	3	*4*	6	8	10	20	100
N_T	13131	9534	9293	*8412*	7169	6231	5846	3835	1822
N_C	742	2220	3430	*3538*	3184	2740	2542	1512	581
N_R	45	2301	3600	*3721*	3186	2832	2394	1599	562
N_{RC}	33	2209	3421	*3532*	3167	2728	2394	1509	562

Table 18: Effectiveness of the planting pattern based descriptor while varying n. All other parameters are kept as indicated in table 13.

D	100	200	300	400	*500*	800	1100	1400	1700
N_T	10032	9289	8905	8655	*8412*	8281	8260	8243	8264
N_C	5693	4833	4286	3866	*3538*	2914	2482	2178	1912
N_R	5965	5079	4508	3630	*3721*	3051	2610	2272	2015
N_{RC}	5643	4811	4208	3569	*3532*	2909	2481	2156	1910

Table 19: Effectiveness of the planting pattern based descriptor while varying D. All other parameters are kept as indicated in table 13. Note that low values of D are possible as ground-truth was used beforehand.

C_1	0.4	0.5	0.6	0.7	*0.8*	0.85	0.9	0.95	1.0
N_T	2106	3353	4422	6005	*8412*	9548	11345	12520	14790
N_C	1621	2268	2656	3087	*3538*	3691	3881	3945	4168
N_R	1615	2355	2774	3192	*3721*	3492	3687	4146	4345
N_{RC}	1535	2265	2630	3027	*3532*	3301	3612	3935	4140

Table 20: Effectiveness of the planting pattern based descriptor while varying C_1. All other parameters are kept as indicated in table 13.

C_1	*0.8*	0.85	0.9	0.95	1.0
C_2	*1.0*	0.95	0.9	0.85	0.8
N_T	*8412*	8020	8305	7403	7539
N_C	*3538*	3134	2895	2396	2206
N_R	*3721*	3301	2564	2516	2203
N_{RC}	*3532*	3127	2444	2389	2104

Table 21: Effectiveness of the planting pattern based descriptor while varying C_1 and C_2 simultaneously. All other parameters are kept as indicated in table 13.

ε	1	2	5	8	*10*	50	100	200	500
N_T	8412	8412	8412	8412	*8412*	8412	8412	8412	8412
N_C	3538	3538	3538	3538	*3538*	3538	3538	3538	3538
N_R	1142	2676	3670	3718	*3721*	3814	4094	4674	7230
N_{RC}	1140	2659	3485	3530	*3532*	3538	3538	3538	3223

Table 22: Effectiveness of the planting pattern based descriptor while varying ε. All other parameters are kept as indicated in table 13.

5.3 Matching and Warping

In this section the *full* method, that is, matching of descriptors as well as correcting one of the orthophotos by is tested without first leveraging ground-truth. The default parameter settings are used for both methods, which are given in tables 2 and 13. After the methods are run, the error is found using RANSAC model fitting, and orthophoto B is corrected by $\mathbf{x}^{B_c} = \mathbf{t} + R\mathbf{x}^B + \mathbf{w}(\mathbf{x}^B)$. Now, the effectiveness of the method is assessed by comparing the median distance $\|\mathbf{p}_i - \mathbf{p}_\iota\|$ between M white cards for $\mathbf{p}_i$ in orthophoto A corresponding to $\mathbf{p}_\iota$ in orthophoto B_c, which should ideally yield zero.

The results while using the shape and size based descriptor, as given in (35) and (38), are given in table 23. The results for the planting pattern descriptor, as in (39), are given in table 24.

A	05-18	05-21	05-24	05-29	06-07	06-12
B	05-21	05-24	05-29	06-07	06-12	06-19
M	878	696	685	1	9	0
median($\|\mathbf{p}_i - \mathbf{p}_\iota\|_2$)	74.9	2.5	95.2	114.2	140.3	n.a.

Table 23: Effectiveness of the full method using the shape and size based descriptor, judged by the median error in centimetres, defined as median($\|\mathbf{p}_i - \mathbf{p}_\iota\|_2$) for M pairs of $(i \leftrightarrow \iota)$. All parameters, excluding the date, are given in table 2. Compare also with the descriptor matching results in table 3. Note the low number of matches in certain columns, which is presumably caused by a poor correction of B_c, hindering the identification of $(i \leftrightarrow \iota)$ pairs.

A	05-18	05-21	05-24	05-29	06-07	06-12
B	05-21	05-24	05-29	06-07	06-12	06-19
M	595	697	645	920	772	1178
median($\|\mathbf{p}_i - \mathbf{p}_\iota\|_2$)	118.4	2.4	88.2	88.6	74.1	64.3

Table 24: Effectiveness of the full method using the planting pattern based descriptor, judged by the median error in centimetres, defined as median($\|\mathbf{p}_i - \mathbf{p}_\iota\|_2$) for M pairs of $(i \leftrightarrow \iota)$. All parameters, excluding the date, are given in table 13. Compare also with the descriptor matching results in table 14.

6 DISCUSSION & CONCLUSION

In this section some of the results will be presented and discussed. First, results that relate to the different parameters associated with each method will be discussed. Next, the method as a whole, as well as its effectiveness on different stages of growth in the time-series, is discussed.

6.1 Parameter Settings

Regarding the shape and size based descriptor, the first parameter is found in n_f, defined by $n = \lfloor n_f L \rfloor$, which governs the sampling of a contour. Based on table 7 it can be observed that choosing $n_f > 1$ is desirable to make optimal use of the given resolution. For the same reason, increasing $n_f \gg 1$ should not yield increased performance. The second parameter is w, defining the number of steps in (37). Based on table 8 a surprisingly low resolution seems to suffice, however, one may easily choose $w = 255$ as this is still very computationally efficient.
Regarding the planting pattern based descriptor, the most important parameter to set is n, as defined in (39). Based on table 18, the choice $n = 4$ seems to be optimal. This may be explained by the fact that the mean distance increases while increasing n, however, the variance increases slower in comparison. In other words, the first so many nearest neighbours have a higher variance over mean distance, σ/μ, presumably giving them more discriminative power, see figure 12.

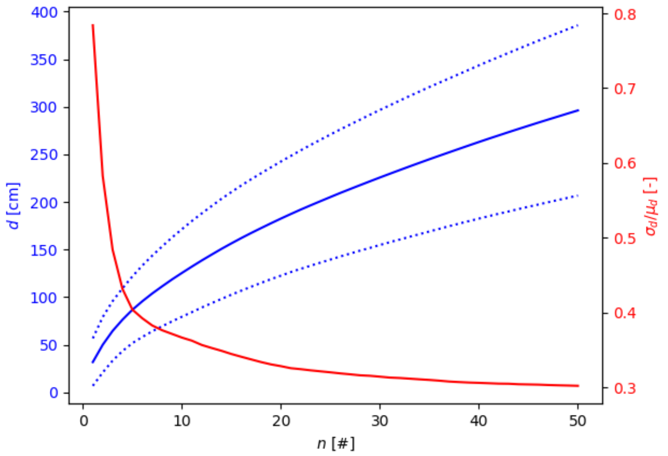


Figure 12: Mean distance plus and minus one standard deviation in centimetres (d, blue) for every n^{th} neighbour up to $n = 50$, as well as the standard deviation over mean (σ/μ, red).

Regarding the matching that is used for both methods, the most important parameter choice is to be made in C_1, as in (55), used to assure the quality of a match. Note that C_1 provides, as it is, a trade-off between the total number of matches N_T, and the percentage of correct matches N_C/N_T. This phenomenon can be observed in figure 13 which is based on table 10. Additionally, the same trade-off is present in table 20.

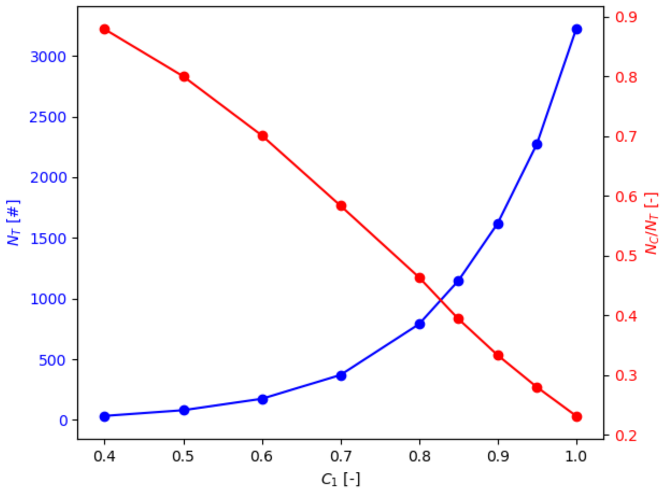


Figure 13: Effect on the total number of matches (N_T, blue) and ratio of correct matches (N_C/N_T, red) of varying the C_1 parameter. Based on data in table 10.

Ideally, one of course has a large number of matches with a high percentage of correct matches, leading to the compromising choice of $C_1 = 0.8$. Note that this value is also used by Lowe (2004) for the scale invariant feature transform.
Additionally, a backwards quality check was presented in (56) with the addition of C_2. Using this, by setting $C_2 < 1$, however, seems to have no positive effect on the method, see tables 11 and 21.

6.2 Time-Series Results

In general, it is concluded that the success of matching, i.e. the chance that RANSAC is able to select all the correct matches and no other matches, is dependent mostly on the percentage of correct matches N_C/N_T. In an attempt to quantify this, it seems that when more than 15 % of the matches is correct, the procedure succeeds. There are, however, exceptions on both sides, for example in table 16 using $A_{min} = 1$ or in table 3 using $A =$ 05-24 and $B =$ 05-29.
Of course, besides the percentage of correct matches, the total number of matches N_T is crucial as well. When comparing table 23 with 3, or 24 with 14, it can also be observed that having a low N_T leads to a poor correction presumably because only parts of the orthophoto are corrected, while in other parts a large error remains.
When considering the success of the method for different steps in the time-series in tables 3, 14, 23, and 24, it can be observed that both methods generate good results for 05-21 to 05-24, indicating that both methods do indeed offer time-invariance to a certain extent. However, both methods perform rather poorly in later growth stages as well as the very early growth stages.
The latter phenomenon may be explained by limiting resolution. Consider figure 10, on 05-18 crops have a diameter roughly between five and ten centimetres, and thus pixels. Consequently, stable detection of the crops, as well as meaningful description of shape and size, is significantly hindered. Presumably, if resolution were to be increased, for example by flying at a lower altitude, the methods may both be successful in earlier growth stages.
The reason both methods perform poorly in later growth stages, however, is most likely caused by the 'merging' of crop canopies. Consider again figure 10, when going from 05-24 to 05-29 it can be observed that some crops begin to grow to a size where -in nadir view- they start forming connected components with other crops. Being able to identify every single crop in each orthophoto, as well as being able to find the contour as in (15) corresponding to a single crop, is crucial to

both suggested methods. Merging of the crop canopies makes this task extremely difficult, as the current implementation relies on dynamic image thresholding to provide this input. Presumably, the stability may be improved by post-processing. For example, one may attempt to split such connected components using shrinkage and buffer operations (Davies, 2018). Another option is to account for mergers in the growth model and descriptors. One may track distances between centroids, as well as the radii of crops, and predict mergers, which could be accounted for in the pattern descriptor in (39). Alternatively, input may be created via a different method altogether using convolutional neural networks. For example, the *YOLOv3* (Redmon and Farhadi, 2018) algorithm is capable of creating bounding boxes for individual crops even if plants have overlapping leafs (E. Verhoeff, personal communication, June 29, 2020). Additionally, using such algorithms may already be desired anyways for agricultural change detection analyses.

REFERENCES

Davies, E. R. (2018). Binary shape analysis. In *Computer Vision: Principles, Algorithms, Applications, Learning*, pages 203–238. Elsevier, 5^{th} edition.

Fathian, K., Ramirez-Paredes, J. P., Doucette, E. A., Curtis, J. W., and Gans, N. R. (2018). QuEst: A quaternion-based approach for camera motion estimation from minimal feature points. *IEEE Robotics and Automation Letters*, 3(2):857–864.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Hartley, R. I. (1997). In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593.

Li, Y., Wang, S., Tian, Q., and Ding, X. (2015). A survey of recent advances in visual feature detection. *Neurocomputing*, 149:736 – 751.

Longuet-Higgins, H. C. (1981). A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770.

Redmon, J. and Farhadi, A. (2018). YOLOv3: An incremental improvement.

Rey-Otero, I. and Delbracio, M. (2014). Anatomy of the SIFT method. *Image Processing On Line*, 4:370–396.