Quantifying Population Movement Using a Novel Implementation of Digital Image Correlation in the ICvectorfields Package

by Devin W. Goodsman

Abstract Movements in imagery captivate the human eye and imagination. They are also of interest in variety of scientific disciplines that study spatiotemporal dynamics. Popular methods for quantifying movement in imagery include particle image velocimetry and digital image correlation. Both methods are widely applied in engineering and materials science, but less applied in other disciplines. This paper describes an implementation of a basic digital image correlation algorithm in R as well as an extension designed to quantify persistent movement velocities in sequences of three or more images. Algorithms are applied in the novel arena of landscape ecology to quantify population movement and to produce vector fields for easy visualization of complex movement patterns across space. Functions to facilitate analyses are available in **ICvectorfields** (Goodsman, 2021). These methods and functions are likely to produce novel insights in theoretical and landscape ecology because they facilitate visualization and comparison of theoretical and observed data in complex and heterogeneous environments.

1 Introduction

Living organisms move through space in complex ways that have inspired many branches of spatial pattern analysis from Turing instabilities (Alonso, Bartumeus, and Catalan 2002; Ruan 1998), to complex systems analysis of the emergent properties of individual-level behaviours when organisms live in groups (Parrish and Edelstein-Keshet 1999; N. Johnson 2009). Moreover, in mathematical ecology there is a long history of deriving analytic expressions for traveling wave speeds from mathematical models of biological systems based on partial differential equations Skellam (1951) and integrodifference equations (Kot, Lewis, and Driessche 1996). In addition to standard traveling waves and wave-trains, simulation studies have revealed more unusual patterns of population level movement can arise from the way organisms interact and move on the landscape (Hassell, Comins, and May 1994). Spiral waves are one example of surprising spatiotemporal dynamics that can arise in biological systems (Hassell, Comins, and Mayt 1991).

Travelling waves and spiral waves emerge from mathematical models of population expansion, which are often based on partial differential equations and integrodifference equations. These types of models, which represent the movement patterns of populations of organisms, are sometimes classified as Eulerian approaches to distinguish them from Lagrangian approaches that focus on the trajectories of individuals. The majority of R packages that quantify organismal movement, however, are Lagrangian as they pertain to the analysis of the tracks or trajectories of individual animals with tracking collars or tracking devices. Integrated step selection models (Avgar et al. 2016), such as those in the **amt** package (Signer, Fieberg, and Avgar 2019), which incorporate the impact of spatially variable habitat or environmental variables on movement of individuals modeled using a discrete time and discrete space framework, are an example of a Lagrangian approach when fitted to movement data from individuals. Because my focus in this work is on population-level movements that are evident in imagery, I will forego further discussion of Lagrangian models and instead refer the interested reader to a review of R packages for modeling animal movement (Joo et al. 2020). At the time of writing, R packages that focus on the Eulerian approach include **IDE** (Zammit-Mangion 2019), **deSolve** (Soetaert, Petzoldt, and Setzer 2010), and **ReacTran** (Soetaert and Meysman 2012). These packages are designed primarily to obtain numerical solutions to Eulerian models, analyze their dynamics, and fit them to data. Recently, movement modeling based on stochastic differential equations, stochastic partial differential equations (Krainski et al. 2018), and other stochastic process models (Buderman et al. 2016) has proliferated. Computationally efficient Bayesian statistical approaches are often required to fit these stochastic models to data due to the ubiquity of noise in spatiotemporal time series in combination with nonlinear dynamical processes (Krainski et al. 2018).

In contrast to the R packages and approaches I have cited above, this work is focused on the description of an empirical method for quantifying spatially heterogeneous rates of movement in sequences of images without fitting a model–although I do approach the quantification problem from an Eulerian perspective. Empirically quantifying spread rates without imposing a specific mathematical model allows the user to abandon many of the assumptions implicit in mathematical modeling of population expansion. For example, tractable mathematical models of consumer-resource

systems that generate traveling waves, wave-trains, and spiral waves, often rely on assumptions of a homogeneous spatial environment with respect to resources or one in which there are no discontinuities in resources. In contrast, many organisms spread in environments that are spatially heterogeneous (Urban et al. 2008), and in environments subject to persistent directional flows that impact organism movement (Hoffman et al. 2006). The ramifications of this claim are more easily understood using a meteorological analogy. In meteorology, vector fields are frequently used to illustrate the impacts of high and low pressure systems on wind speed and direction, and thus on the movement of weather systems. In such meteorological systems, wind speed and direction are complicated functions of topography and complex atmospheric dynamics. As a result, vector fields representing movement in such systems are often variable at the regional scale, with winds flowing in one direction on one side of a map and possibly in an opposing direction on the other side. In ecology, populations of organisms are like the weather systems in that their movement on the landscape is what is of primary interest to researchers; variable wind causes spatially variable movement of weather systems similarly to how persistent directional fluid flows, including wind, in an organism's environment impact dispersal, and therefore population movement.

At the time of writing, the only tool in R (R Core Team 2021) designed to empirically estimate spreading speed or the speed of wave-trains in populations without fitting a mathematical model is implemented in the **ncf** R package (Bjornstad 2020). The **ncf** package relies on lagged non-parametric cross-correlation functions to estimate spreading speed of traveling waves (Bjørnstad and Falck 2001). To do so, it takes two spatiotemporal data sets that differ from one another only in that one is a time-lagged version of the other, and projects their planar coordinates onto lines of varying angles that can be specified using function arguments. After projection onto a line, cross-correlation is estimated using a spline-correlogram approach (Bjørnstad and Falck 2001) and the location of maximum cross-correlation gives an estimate of displacement along the direction of the projection line. This approach was used to estimate the velocity of traveling wave-trains in the larch budmoth system in the European Alps (Bjørnstad et al. 2002).

Projecting population data from a domain with two spatial coordinates onto a domain that has only one spatial coordinate and then using a correlogram approach precludes quantification of more complicated patterns of movement on the landscape. For example if two spatially separated populations are moving towards one another at the same speed, such an approach will yield enigmatic correlograms. Similarly, if several populations move radially around a central fulcrum, the correlogram will be difficult to translate to an estimate of directional movement.

In this paper I present an approach for estimating vector-fields in systems with spatially variable movement that is inspired by a technique from engineering and materials science called Digital Image Correlation or DIC (Anuta 1970; Sutton, Orteu, and Schreier 2009). Among other things, Digital Image Correlation is used to estimate displacement based on photographs of a planar material before, during, and after a force has been applied to warp its surface (Sutton, Orteu, and Schreier 2009). A typical DIC approach as well as the extensions described in this paper are implemented in the **ICvectorfields** package (Goodsman 2021), in which the IC is the abbreviation for Image Correlation. I demonstrate these approaches using the **ICvectorfields** package to analyze a simulated data set as well as the larch budmoth data set provided with the **ncf** R package (Bjornstad 2020; Bjørnstad et al. 2002).

2   Mathematical and Computational Details

Here I provide mathematical and computational details of the algorithms used in the **ICvectorfields** R package starting with a standard digital image correlation approach, and following with extensions to estimate persistent movement and to account for spatial variability in persistent movement. The **ICvectorfields** package capitalizes on the algorithms written in C under the title FFTW which stands for Fastest Fourier Transform in the West (Frigo and Johnson, 2005), and a convenient wrapper package in R called **fftwtools** (Rahim 2021). Input raster images and raster stacks are read and manipulated using the **terra** package (Hijmans 2021).

Digital image correlation

One of the earliest applications of cross-correlation in image analysis was to align images taken from different sensors or at different times using satellites or aircraft (Anuta 1970). The theoretical and computational details I present here loosely follow those in this pioneering application. I will provide the mathematical underpinning of two-dimensional cross-correlation, and then elaborate on its computational implementation, which involves some additional complexity due to the circular nature of discrete fast Fourier transforms. In all descriptions below, I do not normalize the cross-correlation function to obtain Pearson correlation coefficients and therefore, I follow the convention of using the

terms cross-correlation and cross-covariance interchangeably.

Given two images that have been converted to square matrices f and g of dimension $m \times m$, two-dimensional cross-correlation can be defined in terms of a convolution:

$$(f \star g)\left(x_j, y_i\right) = \left(\overline{f\left(-x_j, -y_i\right)} * g\left(x_j, y_i\right)\right)\left(x_j, y_i\right), \tag{1}$$

in which $(f \star g)$ is the two-dimensional cross-correlation matrix, the $*$ operator denotes convolution, $\overline{f\left(-x_j, -y_i\right)}$ is the complex conjugate of the $f\left(x_j, y_i\right)$ matrix, i is the matrix row index, and j is the matrix column index $i, j \in \mathbb{N} = \{1, 2, ...\}$. Note that I use array indices that start at one rather than zero. The coordinates of the centroids of each pixel are given by x_j and y_i.

Based on the convolution theorem, equation (1) can be rewritten as

$$(f \star g)\left(x_j, y_i\right) = \mathbb{F}^{-1}\left(\overline{\mathbb{F}\left(f\left(x_j, y_i\right)\right)}\mathbb{F}\left(g\left(x_j, y_i\right)\right)\right)\left(x_j, y_i\right), \tag{2}$$

wherein $\mathbb{F}$ denotes the two-dimensional Fourier transform, $\mathbb{F}^{-1}$ denotes its inverse, and $\overline{\mathbb{F}\left(f\left(x_j, y_i\right)\right)}$ is the complex conjugate of $\mathbb{F}\left(f\left(x_j, y_i\right)\right)$. Because $\overline{\mathbb{F}\left(f\left(x_j, y_i\right)\right)} = \mathbb{F}\left(\overline{f\left(-x_j, -y_i\right)}\right)$, and because $f\left(x_j, y_i\right)$ contains only real numbers, the complex conjugate can be calculated using matrix multiplication:

$$\overline{f\left(-x_j, -y_i\right)} = r \times f \times r, \tag{3}$$

in which the r matrix is a $m \times m$ matrix that has zeros everywhere except for along the diagonal that runs from its lower left to upper right corners, which contains ones:

$$r = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{pmatrix}. \tag{4}$$

The matrix calculations in equations (3) and (4) substitute for calculation of the complex conjugate only in the case where all values of the f matrix are real, which is the case in most natural science applications.

Together, equations (1) through (4) constitute an elegant way to compute two dimensional cross-correlation. Computer implementation of these, however, requires some additional complexity due to the use of discrete fast Fourier transforms to efficiently compute convolutions. Discrete fast Fourier transforms are inherently circular, which means that what happens on the outer edges of matrices will impact their discrete fast Fourier transform on the opposite side. In order to mitigate this problem, zeros are added to the outer edge on all sides of both the f and g matrices (Anuta 1970). In **ICvectorfields**, the f and g matrices are padded with as many zeros as there are rows and columns in the original matrix and then additional zeros are added to ensure that both matrices are square with an even number of rows and columns.

When matrices are padded as described above and discrete fast Fourier transforms are used as in equation (2), the non-cyclic components of the convolution end up in the outer corners of the cross-correlation matrix $(f \star g)\left(x_j, y_i\right)$ (Anuta 1970). Thus, to obtain a correct estimate of cross-correlation, $(f \star g)\left(x_j, y_i\right)$ must be divided into four quadrants and each quadrant must be flipped along its horizontal and vertical axes using matrix multiplication. For example, if the zero-padded matrices have dimension $k \times k$, each quadrant of $(f \star g)\left(x_j, y_i\right)$ will have dimension $k/2 \times k/2$ and the following operation flips each quadrant matrix:

$$q_f = r \times q \times r, \tag{5}$$

where r is a $k/2 \times k/2$ matrix as in equation (4). Then the four quadrants can be reassembled into the $k \times k$ cross-correlation matrix. The mathematical operations in equations (1) through (5) are implemented in the *Xcov2D* function in the **ICvectorfields** package.

Once a cross-correlation matrix has been estimated, it can be used to compute displacement in the

horizontal and vertical directions in terms of the horizontal and vertical shifts in pixel numbers that maximize cross-correlation. In **ICvectorfields**, shifts to the right and up are designated as positive, whereas shifts to the left and down are considered negative.

A typical implementation of DIC will define a region of interest within the input images or their corresponding matrices wherein displacement vectors are sought using a bounding box as in the *DispFieldbb* function in **ICvectorfields** or using a sub-grid of equal sized regions of interest as in the *DispField* function in **ICvectorfields**. Note that all of the functions in **ICvectorfields** that use DIC or variations of it, translate displacement or velocities in terms of pixel shifts to the spatial units defined in the projection information of the original input rasters. The coordinate information required for translation of pixel shifts to the correct spatial units is obtained using functions in the **terra** R package (Hijmans 2021).

Extending DIC to quantify persistent movement

In applications of DIC in earth systems with persistent directional flows that influence movement, it is valuable to determine directional movement of populations of interest that persist for more than one time step. In such situations, a spatiotemporal array of images with two space dimensions and one time dimension is required. Often these can be formulated as stacks of raster images, with each layer in the stack representing spatially referenced observations for one time step (step one in Fig. 1). A variation of DIC which I call Spatiotemporal Image Correlation (STIC) permits estimation of persistent directional movement in terms of orthogonal velocity vectors.

In STIC, the three dimensional array is first lagged by duplicating it and then removing an integer number of layers from the top of one duplicate and the bottom of the other (steps two and three in Fig. 1. The integer lag is user defined and serves to minimize estimates of zero movement which always would occur in the absence of a lag. To differentiate the duplicate lagged arrays, I will refer to the first as the reference array, and the second as the lagged array. Regions of interest in the reference array are selected and locations outside the region of interest in the reference array are assigned values of zero (grey shaded region in steps two and three in Fig. 1 represent regions of interest). The reference array and the lagged array are then dimension reduced by averaging along rows to obtain one pair of two-dimensional matrices and by averaging along columns to obtain a second pair of two dimensional matrices (step 4 in Fig. 1). The first pair of matrices comprises row-averaged reference and lagged matrices. The second pair of matrices comprises column-averaged reference and lagged matrices. Each matrix in the two pairs has one space dimension and one time dimension.

Cross-correlation between the pairs of reference and lagged matrices is then computed as described for DIC. Recall that one dimension of each of the row or column-averaged matrices is spatial while the other is temporal, which enables calculation of two orthogonal velocity vectors based on space shifts and time shifts obtained by application of DIC:

$$v_x = s_x/s_{tx}, s_{tx} \neq 0 \qquad (6)$$
$$v_y = s_y/s_{ty}, s_{ty} \neq 0 \qquad (7)$$

in which v_x and v_y are velocity in the horizontal and vertical directions, s_x and s_y are shifts in the horizontal and vertical direction, s_{tx} is the time shift that corresponds to spatial shifts in the horizontal direction, and s_{ty} is the time shift that corresponds to spatial shifts in the vertical direction. Note that due to the time shift, the user-defined time lag does not necessarily pre-determine the denominator of the orthogonal velocity vectors.

Spatially variable velocities

When the magnitudes of movement velocities are highly spatially variable, a single time lag is not optimal for quantifying orthogonal velocity vectors. For these scenarios a variation on the STIC algorithm called STIC+ allows the user to specify a maximum time lag. The algorithm then repeats the steps described for STIC for each integer time lag from one to the maximum time lag. For each repetition and each location of interest, the total velocity magnitude (speed) is calculated as

$$|v| = \sqrt{v_x^2 + v_y^2}. \qquad (8)$$

For each region of interest, the horizontal and vertical velocity vectors are determined by the time lag STIC calculation that maximizes equation (8).

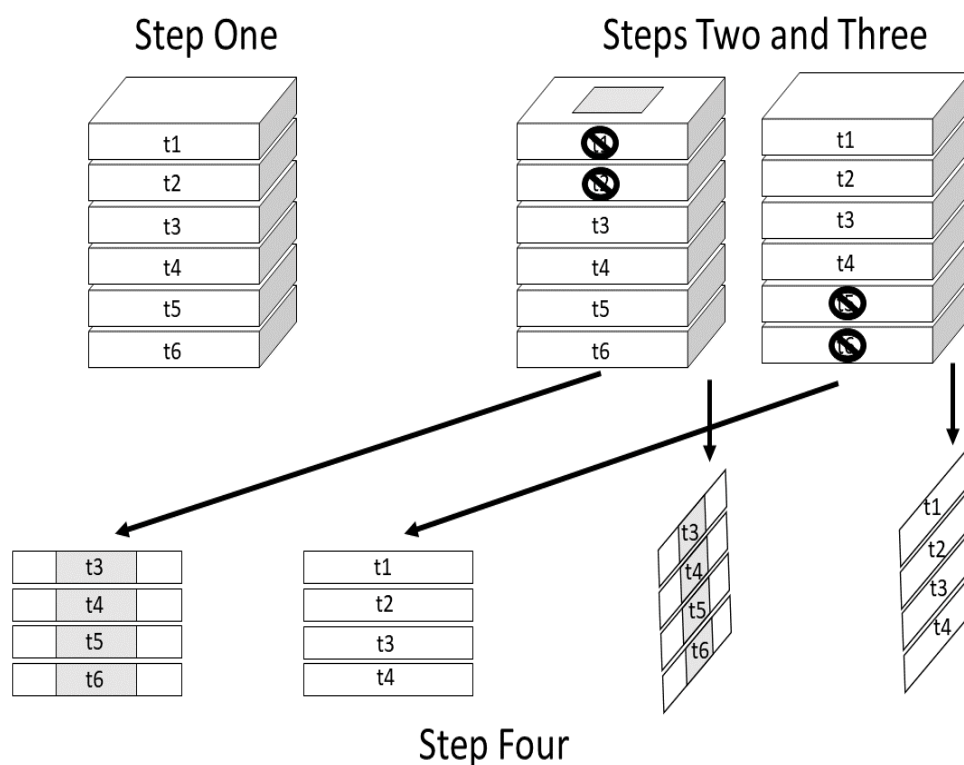A summary table describing which functions in **ICvectorfields** use each of the algorithms described

Figure 1: The STIC algorithm: The input array in step one is a raster stack of images in which each image layer represents the phenomenon of interest in planar space at a different time instance. In step two, the input array is duplicated and based on a user specified time lag, layers are removed from the top of one array and the bottom of the other. In addition, a region of interest is defined in one of the duplicate arrays represented by the grey shaded region at the top of the prism on the left. In step four the rows are dimension reduced by averaging along one of the axes (either rows or columns). This produces a pair or row-averaged matrices and a pair of column-averaged matrices that are analyzed using cross-correlation to estimate orthogonal velocity vectors.

Table 1: ICvectorfields functions, algorithms, and use contexts to facilitate decisions on which function is most applicable. ROI stands for region of interest, which is defined either using a grid or a bounding box, Velocities refers to whether the magnitudes of velocities in the vector field are presumed to be spatially variable or not.

function	algorithm	images	ROI	velocities
DispField	DIC	2	grid	variable or not
DispFieldbb	DIC	2	bounding box	variable or not
DispFieldST	STIC	3+	grid	less variable
DispFieldSTbb	STIC	3+	bounding box	less variable
DispFieldSTall	STIC+	3+	grid	more variable
DispFieldSTbball	STIC+	3+	bounding box	more variable

above is provided (Table 1). Two functions in **ICvectorfields** use a standard implementation of DIC similar to that described by (Anuta 1970), two functions use the STIC extension and two functions use the STIC+ extension (Table 1).

3 Application

In this section I demonstrate the use of DIC and extensions implemented in **ICvectorfields** (Goodsman 2021) using an example in which data were simulated based on a partial differential equation and using the classic larch budmoth defoliation data that are embedded in the **ncf** package (Bjornstad 2020). The data from the simulated example are embedded in **ICvectorfields**. For visualization of results, the demonstrations call functions in several R packages: These include **ggplot2** (Wickham 2016), **ggnewscale** (Campitelli 2021a), **metR** (Campitelli 2021b), and **terra** (Hijmans 2021).

Demonstration using simulated data

The model used to simulate data to test and demonstrate the functionality of **ICvectorfields** is a convection-diffusion equation, which is a partial differential equation with terms for diffusion, advection, and reaction:

$$\frac{\partial u}{\partial t} = \nabla \cdot (D \nabla u) - \nabla \cdot (\mathbf{v} u) + ru, \tag{9}$$

in which r is the growth rate with units of per unit time, D is the diffusion coefficient with squared spatial units per unit time, $\mathbf{v}$ is the bivariate orthogonal velocity vector in units of space per unit time, ∇ represents the gradient, and $\nabla \cdot$ represents divergence. The orthogonal velocity vector is spatially dependent in the simulations that follow:

$$\mathbf{v} = \begin{cases} (0, 0.2), & x < 0, y \geq 0 \\ (0.2, 0), & x \geq 0, y \geq 0 \\ (0, -0.2), & x \geq 0, y < 0 \\ (-0.2, 0), & x < 0, y < 0 \end{cases} \tag{10}$$

Note that by convention in equations (9) and (10), movement to the right and up has a negative sign, and movement to the left and down has a positive sign. This is the opposite convention used in **ICvectorfields**. Note that the discontinuities at $x = 0$ and $y = 0$ in the advection term in equation (10) create strange model behaviours once concentration reaches $x = 0$ or $y = 0$, and so the model was simulated for only 6 time steps to minimize encounters with these axes. Parameter values for the diffusion coefficient and the growth rate were $D = (0.01, 0.01)$ squared spatial units per unit time and $r = 0.5$ per unit time.

The model in equations (9) and (10) was simulated using the **ReacTran** R package (Soetaert and Meysman 2012), using a finite differencing scheme with backward differencing on a square domain of 202 cells in each direction, each with a width of 0.049 spatial units. The initial condition was a concentration of one units per arbitrary unit of volume in the centre of each quadrant of the spatial domain. Boundary conditions were zero flux (reflecting) on all four sides of the spatial domain. The simulation data are saved in table format within **ICvectorfields**.

The data are imported and then converted from table format to a raster stack using the *RastStack-Data* helper function. They can then be visualized as rasters as shown in Fig. 2.
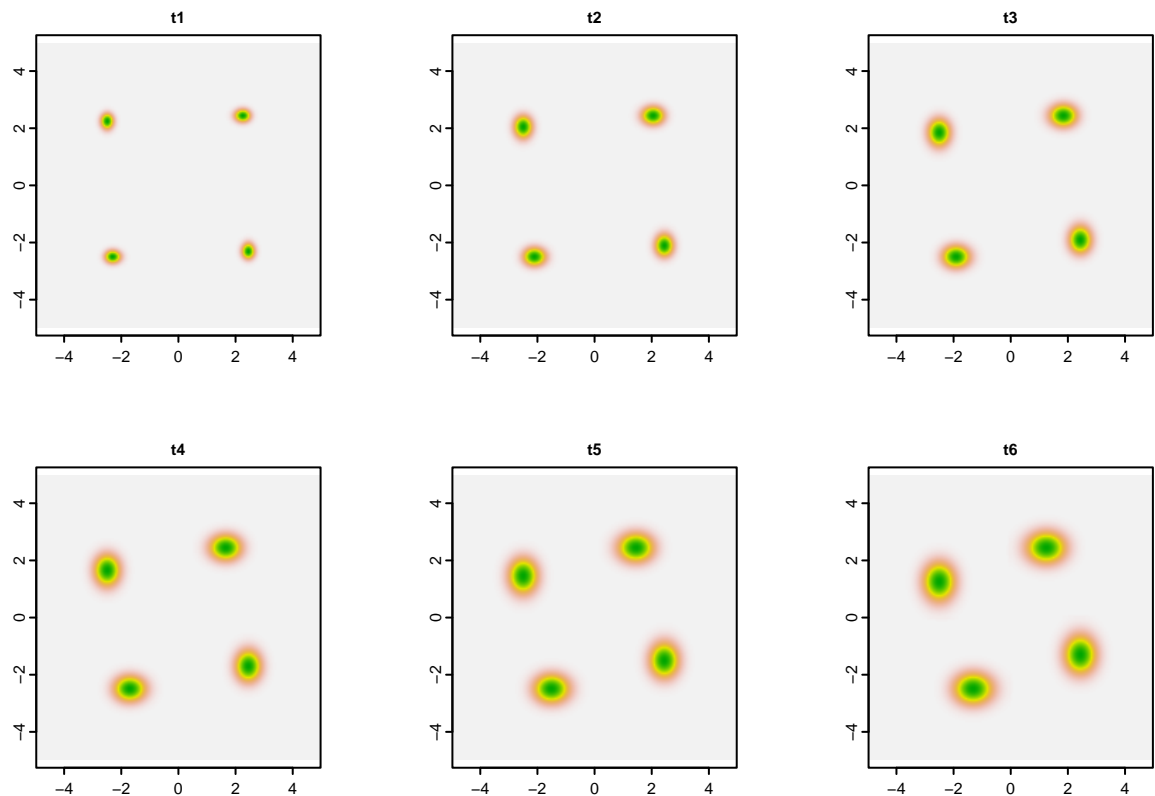
Figure 2: Visualization of simulation data for six time steps. The initial condition at t0 is not shown. Green colours represent the highest concentrations. The populations in each quadrant move counter-clockwise as time steps forward.

```
# import simulated data
data(SimData, package = "ICvectorfields")

# convert to raster stack
SimStack <- ICvectorfields::RastStackData(SimData)

# confirming dimension
#dim(SimStack)

# plotting
layout(matrix(1:6, 2, 3, byrow = TRUE))
#layout.show(6)
terra::plot(SimStack[[1]], legend = FALSE, main = "t1")
terra::plot(SimStack[[2]], legend = FALSE, main = "t2")
terra::plot(SimStack[[3]], legend = FALSE, main = "t3")
terra::plot(SimStack[[4]], legend = FALSE, main = "t4")
terra::plot(SimStack[[5]], legend = FALSE, main = "t5")
terra::plot(SimStack[[6]], legend = FALSE, main = "t6")
```

To analyze displacement based on a pair of images, I chose to use the standard implementation of DIC in the *DispField* function of **ICvectorfields**. The first two arguments of the *DispField* function are the input rasters. The first input raster is treated as the reference image and the second is treated as the shifted image. In this case, the first image is the raster layer corresponding to the first time step (t1) and the second image is the raster layer corresponding to the sixth time step (t6). The function selects regions of interests based on a grid of dimensions given in the *factv1* and *facth1* arguments, which represent to number of rows and columns in each sub-grid. Sub-grids start in the upper left corner and as many sub-grids as fit within the original domain are constructed. In the code below, sub-grids are 101 × 101, which is approximately the size of one quarter of the original spatial domain of the simulation. The restricted argument is by default set to *FALSE*. In that case, the DIC algorithm

Table 2: ICvectorfields output from a call of the DispField function using simulated data. The table is a duplicate of the data table returned after calling the function except that columns 3 through 6 have been omitted so that the table fits within page width limits.

rowcent	colcent	centx	centy	dispx	dispy
51	51	-2.487624	2.487624	0.0000000	-0.9851975
152	51	-2.487624	-2.487624	0.9851975	0.0000000
51	152	2.487624	2.487624	-0.9851975	0.0000000
152	152	2.487624	-2.487624	0.0000000	0.9851975

cross-correlates each region of interest in the first image with the entirety of the second image. When *restricted = TRUE*, the algorithm cross correlates both images only within the region of interest. If the user has reason to believe that movement is predominantly occurring within sub-grids the *restricted = TRUE* option has the added benefit of speeding up computation.

```
# Estimating displacement of simulated data using the DispField function
VFdf1 <- DispField(SimStack[[1]], SimStack[[6]], factv1 = 101, facth1 = 101,
                   restricted = TRUE)
```

The output of *DispField* is in data table format. Because the data table is small, the output is duplicated in Table 2.

The directions of movement coincide with the directions of advection in the simulation with movement downwards in the upper left quadrant (first row of Table 2), movement to the right in the lower left quadrant (second row of Table 2), movement to the left in the upper right quadrant (third row of Table 2), and upwards movement in the lower right quadrant (fourth row of Table 2). Speed of movement can be computed by dividing displacement by the number of time steps that passed $0.98/5 = 0.196$, which is slightly slower than the simulated advection speed of 0.2 spatial units per time step. The discrepancy is likely due to the blurring effect of diffusion in the partial differential equation (equation (9)).

In situations where the speed is constant, velocity can be estimated from pairs of images as I have demonstrated above. However, the *DispFieldST* algorithm is designed to return orthogonal velocity vectors and so for confirmation purposes, I demonstrate it below:

```
# Estimating orthogonal velocity vectors of simulated data using the DispFieldST function
VFdf2 <- DispFieldST(SimStack, lag1 = 1, factv1 = 101, facth1 = 101, restricted = TRUE)
```

The data table that is returned after running the code above looks similar to the data table duplicated in Table **??** except that under the heading dispx and dispy the algorithm returns horizontal and vertical velocities rather than displacement vectors. The directions of movement are the same as those shown in Table 2, but the speed is 0.196 spatial units per unit time as previously estimated.

The vector field and the raw data can be visualized simultaneously using plotting functionality in **ggplot2** with extensions in the **ggnewscale** and **metR** packages.

```
SimVF <- ggplot() +
  xlim(c(-5, 5)) +
  ylim(c(-5, 5)) +
  geom_raster(data = SimData,
              aes(x = xcoord, y = ycoord, fill = t1)) +
  scale_fill_gradient(low = "white", high = "blue", na.value = NA) +
  new_scale("fill") +
  geom_raster(data = SimData,
              aes(x = xcoord, y = ycoord, fill = t6), alpha = 0.5) +
  scale_fill_gradient(low = "white", high = "red", na.value = NA) +
  geom_vector(data = VFdf2,
              aes(x = centx, y = centy,
                  mag = Mag(dispx, dispy),
                  angle = Angle(dispx, dispy))) +
  theme(panel.background = element_rect(fill = "white", color = "grey"),
        legend.key.size = unit(0.4, 'cm'))

SimVF
```
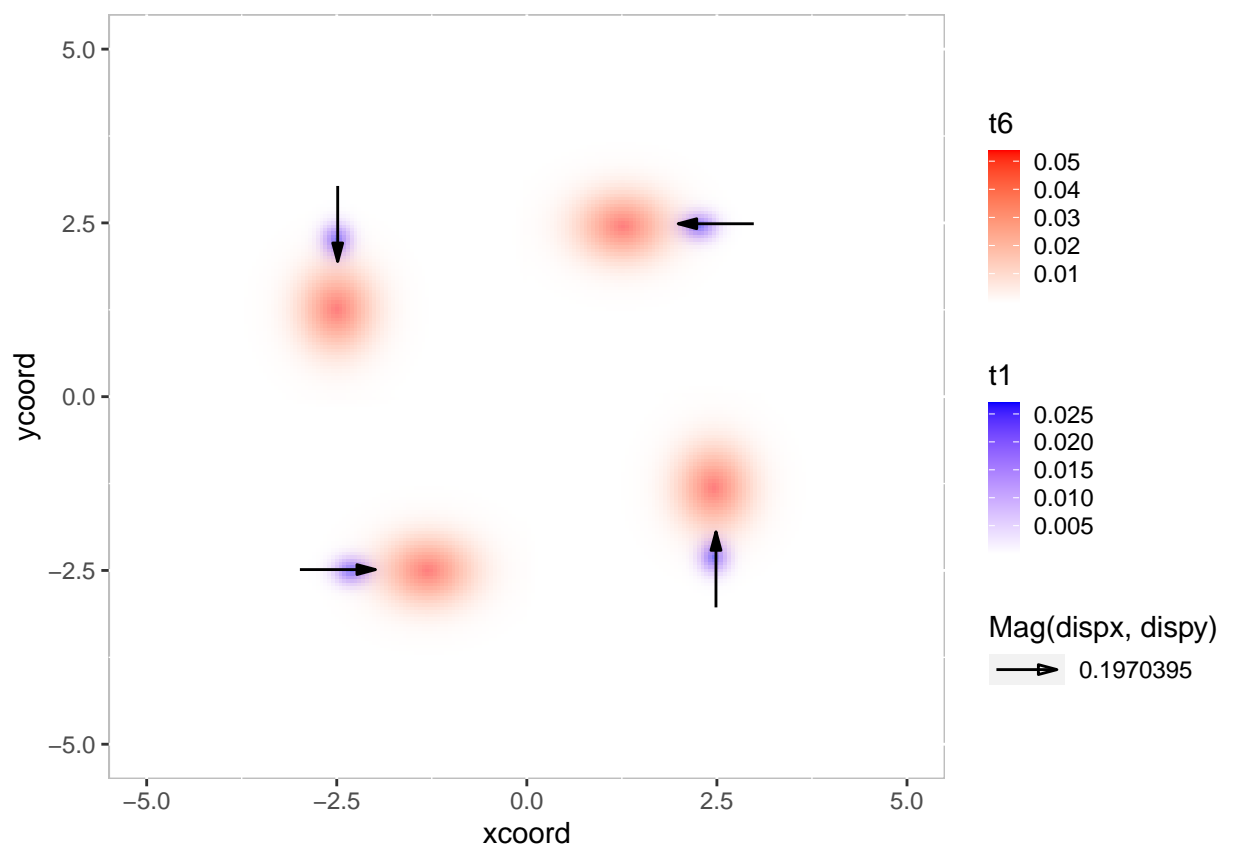
Figure 3: Vector field for radial movement simulated using a convection-diffusion equation. The orthogonal velocity vectors are estimated using the DispFieldST function in the ICvectorfields package.

Table 3: Output from a call of the DispFieldST function using simulated data. This call is meant to demonstrate a potential pitfall in using the cross-correlation approach because when restricted = FALSE, the algorithm finds positive cross-correlations that are not caused by movement. The table is a duplicate of the data table returned after calling the function except that columns 3 through 6 have been omitted so that the table fits within page width limits.

rowcent	colcent	centx	centy	dispx	dispy
51	51	-2.487624	2.487624	0.0985198	-3.8915302
152	51	-2.487624	-2.487624	3.8915302	0.0985198
51	152	2.487624	2.487624	-3.8915302	-0.0985198
152	152	2.487624	-2.487624	-0.0985198	3.8915302

The resulting figure is duplicated in Fig. 3. The velocity vectors in the vector field are consistent with the simulated advection vectors (Fig. 3), although they slightly underestimate movement speed.

Before proceeding to the next demonstration I will illustrate one of the potential pitfalls of estimating movement based on cross-correlation. If the argument of the *DispFieldST* function is left its default *restricted = FALSE* configuration, the algorithm will search the entire domain for shifts that maximize cross-correlation. Because the simulations in each quadrant of the spatial domain are quite similar, cross-correlation is in fact maximized by shifts that cross quadrants, even though simulated movement was not that large. Therefore, calling *DispFieldST* with *restricted = FALSE* produces incorrect output (Table 3): The simulated advection speed is not at all close to the estimated maximum orthogonal advection speed of 3.89 spatial units per unit time.

```
# Estimating orthogonal velocity vectors of simulated data using the DispFieldST function
VFdf3 <- DispFieldST(SimStack, lag1 = 1, factv1 = 101, facth1 = 101, restricted = FALSE)
```

Demonstration using larch budmoth data

Larch budmoths are lepidopteran defoliators that exhibit periodic outbreaks every 8 to 9 years in the European Alps (Bjørnstad et al. 2002). The larch budmoth data originated from survey information collected by the forest administrative agencies of France, Italy, Switzerland, and Austria from 1961 to 1998. The data record the presence of defoliation by larch budmoth caterpillars within 1×1 km grid cells (a binary variable). These data were aggregated up to 20×20 km grid cells so that records at this spatial scale were population proxies for larch budmoth caterpillar abundance (Bjørnstad et al. 2002) based on the assumption that defoliation damage is proportional to the abundance of the causal agents. Grid cells were excluded from the data set if they exhibited less than one percent defoliation or if more than ninety percent of years in which data were collected at that location exhibited no defoliation by larch budmoth (Bjørnstad et al. 2002). The larch budmoth defoliation data exhibit directional traveling wave-trains that travel from the southwest to the northeast along the European Alps (Bjørnstad et al. 2002). These data are embedded in the **ncf** R package.

After loading the **ncf** package as well as **ICvectorfields**, the data can be loaded, converted to a raster stack and visualized as follows:

```
# import larch budmoth data
data(lbm, package = "ncf")

# convert to raster stack
LBMStack <- ICvectorfields::RastStackData(lbm)

# confirming dimension
#dim(LBMStack)

# visualizing
layout(matrix(1:6, 2, 3, byrow = TRUE))
#layout.show(6)
terra::plot(LBMStack[[1]], legend = FALSE, main = "1961")
terra::plot(LBMStack[[2]], legend = FALSE, main = "1962")
terra::plot(LBMStack[[3]], legend = FALSE, main = "1963")
terra::plot(LBMStack[[4]], legend = FALSE, main = "1964")
terra::plot(LBMStack[[5]], legend = FALSE, main = "1965")
```
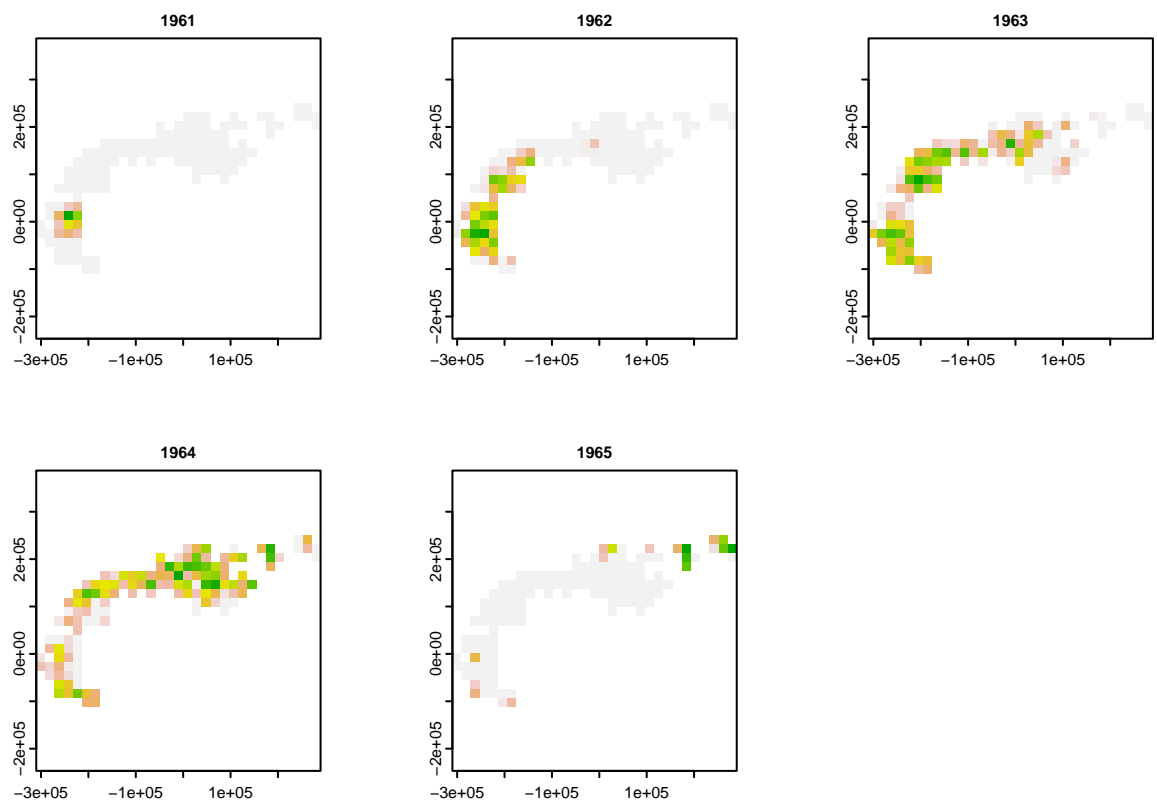
Figure 4: The first five years of the larch budmoth defoliation data included in the ncf package. Green colours represent the highest level of defoliation. Defoliation intensity moves from the southwest to the northeast but not along a straight trajectory between the southwest and northeast corners.
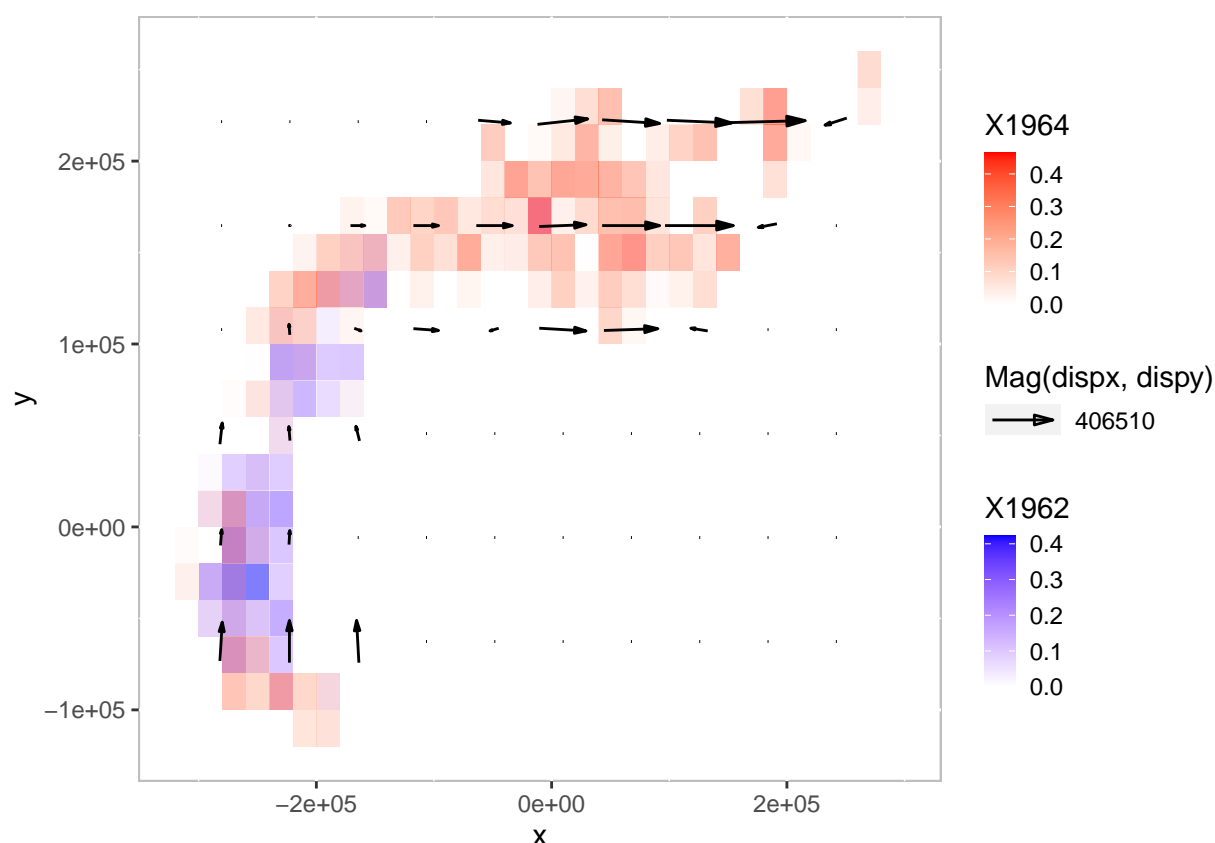
Figure 5: Vector field for Larch Budmoth persistent movement. The orthogonal velocity vectors are estimated using the DispFieldSTall function in the ICvectorfields package. Blue colours show the locations and intensities of defoliation in 1962 and red colours show the locations and intensities of defoliation in 1964. Vectors have their own scale that is distinct from the scale of the map. The figure shows that velocities point predominantly to the north on the southwest corner of the map and predominantly to the east on the northeast corner of the map. In addition, outbreaks appear to slow down as they turn the corner from southwest to northeast.

This code plots the first five years of the data set (Fig. 4), which show a standard progression of outbreaks from the southwest corner of the Alps to the northeast. This pattern repeats relatively regularly every 8 to 10 years in the data set.

The study region covers a large geographic area and so it is likely the population movement speeds vary geographically. For this reason, I elected to use the STIC+ algorithm to analyze the data using *DispFieldSTall*. In the code below I analyze the first 23 years of the time series (1961 to 1983) as defoliation patterns from 1984 to 1998 are less regular.

```
# calculating velocity field for larch budmoth
VFdf3 <- DispFieldSTall(LBMStack[[1:23]], lagmax = 3, factv1 = 3, facth1 = 3, restricted = FALSE)
```

Calling *DispFieldSTall* returns a data frame object that is convenient for plotting the vector field. The vector field reveals that moth movement is to the north on the southwestern side of the Alps and then to the east on the northern side of the Alps (Fig. 5). It also reveals deceleration as outbreaks turn the corner and then acceleration as outbreaks move eastward (Fig. 5).

The average speed of larch budmoth movement can be computed from the data frame output of *DispFieldSTall* as follows:

```
# calculating average speed of population movement
VFdf3$speed <- sqrt((VFdf3$dispx^2) + VFdf3$dispy^2)

# sub-setting to remove locations where speed is zero
VFdf4 <- subset(VFdf3, speed > 0)
```

```
# computing mean, standard deviation and dimension of data frame
# to obtain sample size
mean(VFdf4$speed)

#> [1] 175810

#sd(VFdf4$speed)
#dim(VFdf4)

# upper and lower Wald-type 95 percent confidence interval on average speed
mean(VFdf4$speed)/1000 + qt(0.975, dim(VFdf4)[1] - 1)*sd(VFdf4$speed)/1000/sqrt(dim(VFdf4)[1] - 1)

#> [1] 218.5415

mean(VFdf4$speed)/1000 + qt(0.025, dim(VFdf4)[1] - 1)*sd(VFdf4$speed)/1000/sqrt(dim(VFdf4)[1] - 1)

#> [1] 133.0786
```

Using the approach above, the average movement speed is estimated as 176 ± 43 km$(\text{Yr})^{-1}$, an estimate that is less than the previous speed estimates for northeastern spread of 220 km$(\text{Yr})^{-1}$ (Bjørnstad et al. 2002) and 254 km$(\text{Yr})^{-1}$ (D. M. Johnson, Bjørnstad, and Liebhold 2004). The difference between estimates in the literature and estimates produced here are likely due to the direction of movement. The vectors in the larch budmoth vector field point predominantly north and east (Fig. 5. In other words they are orthogonal. In contrast the movement speeds estimated by Bjørnstad et al. (2002) and D. M. Johnson, Bjørnstad, and Liebhold (2004) are projected along lines that run to the northeast. A simple application of geometry reveals that an average speed of 176 km$(\text{Yr})^{-1}$ in the north and east directions corresponds to an estimated speed of 249 km$(\text{Yr})^{-1}$ in the northeastern direction (Pythagorean theorem $\sqrt{176^2 + 176^2}$). This estimate is consistent with prior speed estimates for larch budmoth population movement (Bjørnstad et al. 2002; D. M. Johnson, Bjørnstad, and Liebhold 2004).

4 Summary

The **ICvectorfields** R package implements standard Digital Image Correlation algorithms in addition to a novel extension that permits estimation of orthogonal velocities of persistent movement in series of three or more images. Here I demonstrate the usefulness of DIC and the extension implemented in **ICvectorfields** in a new arena: Whereas DIC is often applied in engineering and materials science to quantify the effects of force application on materials (Sutton, Orteu, and Schreier 2009), it has not been used in landscape ecology. In this field, the approach has potential to provide new insights into how populations move across landscapes and to demonstrate the untenable nature of assumptions of homogeneity inherent in most analyses based on the traveling wave paradigm. Even when models of sufficient complexity to capture environmental heterogeneity can be used, I expect that the methods in **ICvectorfields** will be useful because they facilitate comparison between modeled and empirical population movement data as demonstrated in the partial differential equation example in this study. Approaches such as this one that estimate movement based on cross-correlation, however, have a weakness: Under certain circumstances, they are prone to finding cross-correlations that are unrelated to movement as was demonstrated in this paper. For this reason, users must exercise vigilance in interpreting the results of vector field analyses like those demonstrated herein. If possible, results should be checked against a standard or against prior published results regarding movement propensity. Nevertheless, the methods described here hold promise for exploratory analyses, hypothesis generation, and synoptic pattern analyses of population movements.

5 Acknowledgements

References

Alonso, David, Frederic Bartumeus, and Jordi Catalan. 2002. "Mutual Interference Between Predators Can Give Rise to Turing Spatial Patterns." *Ecology* 83 (1): 28–34.

Anuta, Paul E. 1970. "Spatial Registration of Multispectral and Multitemporal Digital Imagery Using Fast Fourier Transform Techniques." *IEEE Transactions on Geoscience Electronics* 8 (4): 353–68.

Avgar, Tal, Jonathan R Potts, Mark A Lewis, and Mark S Boyce. 2016. "Integrated Step Selection Analysis: Bridging the Gap Between Resource Selection and Animal Movement." *Methods in Ecology and Evolution* 7 (5): 619–30.

Bjornstad, Ottar N. 2020. *Ncf: Spatial Covariance Functions*. https://CRAN.R-project.org/package=ncf.

Bjørnstad, Ottar N, and Wilhelm Falck. 2001. "Nonparametric Spatial Covariance Functions: Estimation and Testing." *Environmental and Ecological Statistics* 8 (1): 53–70.

Bjørnstad, Ottar N, Mikko Peltonen, Andrew M Liebhold, and Werner Baltensweiler. 2002. "Waves of Larch Budmoth Outbreaks in the European Alps." *Science* 298 (5595): 1020–23.

Buderman, Frances E, Mevin B Hooten, Jacob S Ivan, and Tanya M Shenk. 2016. "A Functional Model for Characterizing Long-Distance Movement Behaviour." *Methods in Ecology and Evolution* 7 (3): 264–73.

Campitelli, Elio. 2021a. *Ggnewscale: Multiple Fill and Colour Scales in 'Ggplot2'*. https://CRAN.R-project.org/package=ggnewscale.

———. 2021b. *metR: Tools for Easier Analysis of Meteorological Fields*. https://doi.org/10.5281/zenodo.2593516.

Goodsman, Devin. 2021. *ICvectorfields: Vector Fields from Spatial Time Series of Population Abundance*. https://CRAN.R-project.org/package=ICvectorfields.

Hassell, Michael P, Hugh N Comins, and Robert M May. 1994. "Species Coexistence and Self-Organizing Spatial Dynamics." *Nature* 370 (6487): 290–92.

Hassell, Michael P, Hugh N Comins, and Robert M Mayt. 1991. "Spatial Structure and Chaos in Insect Population Dynamics." *Nature* 353 (6341): 255–58.

Hijmans, Robert J. 2021. *Terra: Spatial Data Analysis*. https://CRAN.R-project.org/package=terra.

Hoffman, Aaron L, Julian D Olden, Jeremy B Monroe, N LeRoy Poff, Todd Wellnitz, and John A Wiens. 2006. "Current Velocity and Habitat Patchiness Shape Stream Herbivore Movement." *Oikos* 115 (2): 358–68.

Johnson, Derek M, Ottar N Bjørnstad, and Andrew M Liebhold. 2004. "Landscape Geometry and Travelling Waves in the Larch Budmoth." *Ecology Letters* 7 (10): 967–74.

Johnson, Neil. 2009. *Simply Complexity: A Clear Guide to Complexity Theory*. Simon; Schuster.

Joo, Rocio, Matthew E Boone, Thomas A Clay, Samantha C Patrick, Susana Clusella-Trullas, and Mathieu Basille. 2020. "Navigating Through the r Packages for Movement." *Journal of Animal Ecology* 89 (1): 248–67.

Kolmogorov, Andrei N. 1937. "Étude de l'équation de La Diffusion Avec Croissance de La Quantité de Matière Et Son Application à Un Problème Biologique." *Bull. Univ. Moskow, Ser. Internat., Sec. A* 1: 1–25.

Kot, Mark, Mark A Lewis, and Pauline van den Driessche. 1996. "Dispersal Data and the Spread of Invading Organisms." *Ecology* 77 (7): 2027–42.

Krainski, Elias, Virgilio Gómez-Rubio, Haakon Bakka, Amanda Lenzi, Daniela Castro-Camilo, Daniel Simpson, Finn Lindgren, and Håvard Rue. 2018. *Advanced Spatial Modeling with Stochastic Partial Differential Equations Using r and INLA*. Chapman; Hall/CRC.

Parrish, Julia K, and Leah Edelstein-Keshet. 1999. "Complexity, Pattern, and Evolutionary Trade-Offs in Animal Aggregation." *Science* 284 (5411): 99–101.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Rahim, Karim. 2021. *Fftwtools: Wrapper for 'Fftw3' Includes: One-Dimensional, Two-Dimensional, Three-Dimensional, and Multivariate Transforms*. https://CRAN.R-project.org/package=fftwtools.

Ruan, SHIGUI. 1998. "Turing Instability and Travelling Waves in Diffusive Plankton Models with Delayed Nutrient Recycling." *IMA Journal of Applied Mathematics* 61 (1): 15–32.

Signer, Johannes, John Fieberg, and Tal Avgar. 2019. "Animal Movement Tools (Amt): R Package for Managing Tracking Data and Conducting Habitat Selection Analyses." *Ecology and Evolution* 9: 880–90. https://doi.org/10.1002/ece3.4823.

Skellam, John Gordon. 1951. "Random Dispersal in Theoretical Populations." *Biometrika* 38 (1/2): 196–218.

Soetaert, Karline, and Filip Meysman. 2012. "Reactive Transport in Aquatic Ecosystems: Rapid Model Prototyping in the Open Source Software r." *Environmental Modelling & Software* 32: 49–60.

Soetaert, Karline, Thomas Petzoldt, and R. Woodrow Setzer. 2010. "Solving Differential Equations in R: Package deSolve." *Journal of Statistical Software* 33 (9): 1–25. https://doi.org/10.18637/jss.v033.i09.

Sutton, Michael A, Jean Jose Orteu, and Hubert Schreier. 2009. *Image Correlation for Shape, Motion and Deformation Measurements: Basic Concepts, Theory and Applications*. Springer Science & Business Media.

Urban, Mark C, Ben L Phillips, David K Skelly, and Richard Shine. 2008. "A Toad More Traveled: The

Heterogeneous Invasion Dynamics of Cane Toads in Australia." *The American Naturalist* 171 (3): E134–48.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Zammit-Mangion, Andrew. 2019. *IDE: Integro-Difference Equation Spatio-Temporal Models*. https://CRAN.R-project.org/package=IDE.

Bibliography

M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2): 216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation". [p122]

D. Goodsman. *ICvectorfields: Vector Fields from Spatial Time Series of Population Abundance*, 2021. URL https://CRAN.R-project.org/package=ICvectorfields. R package version 0.0.2. [p121]

Devin W. Goodsman
Natural Resources Canada
5320 122 St NW,
Edmonton, Alberta, T6H 3S5
Canada
NA
ORCiD: 0000-0003-1935-5779
devin.goodsman@nrcan-rncan.gc.ca