Astronomy and Computing 10 (2015) 22–31

Contents lists available at ScienceDirect

Astronomy and Computing

journal homepage: www.elsevier.com/locate/ascom

Full length article

FellWalker—A clump identification algorithm☆

D.S. Berry

Joint Astronomy Centre, 660 N. A'ohōkū Place, Hilo, HI 96720, USA

ARTICLE INFO

Article history:
Received 21 July 2014
Accepted 21 November 2014
Available online 17 December 2014

Keywords:
Methods: data analysis
Clump identification
Starlink

ABSTRACT

This paper describes the FellWalker algorithm, a *watershed* algorithm that segments a 1-, 2- or 3-dimensional array of data values into a set of disjoint clumps of emission, each containing a single significant peak. Pixels below a nominated constant data level are assumed to be background pixels and are not assigned to any clump. FellWalker is thus equivalent in purpose to the CLUMPFIND algorithm. However, unlike CLUMPFIND, which segments the array on the basis of a set of evenly-spaced contours and thus uses only a small fraction of the available data values, the FellWalker algorithm is based on a gradient-tracing scheme which uses all available data values. Comparisons of CLUMPFIND and FellWalker using a crowded field of artificial Gaussian clumps, all of equal peak value and width, suggest that the results produced by FellWalker are less dependent on specific parameter settings than are those of CLUMPFIND.

1. Introduction

The CLUMPFIND algorithm (Williams et al., 1994, ascl:1107.014) has been widely used for decomposing 2- and 3-dimensional data into disjoint clumps of emission, each associated with a single significant peak. It is based upon an analysis of a set of evenly spaced contours derived from the data array and has two main parameters—the lowest contour level, below which data is ignored, and the interval between contours. However it has often been noted (*e.g.* Christie et al., in preparation, Kainulainen et al., 2009, Pineda et al., 2009, Smith et al., 2008, Elia et al., 2007 and Brunt et al., 2003) that the decomposition produced by CLUMPFIND can be very sensitive to the specific value used for the contour interval, particularly for 3-dimensional data and crowded fields. The choice of an optimal contour interval is a compromise—real peaks may be missed if the interval is too large, but noise spikes may be interpreted as real peaks if the interval is too small.

The FellWalker algorithm attempts to circumvent these issues by avoiding the use of contours altogether. Only a small fraction of the available pixel values fall on the contour levels used by CLUMPFIND—the majority fall *between* these levels and so will have no effect on the resulting decomposition. By contrast, FellWalker makes equal use of all available pixel values above a stated threshold.

The name "FellWalker" relates to the popular British pass-time of walking up the hills and mountains of northern England, particularly those of the Lake District (see Fig. 1 and http://en.wikipedia.org/wiki/Hillwalking), and was chosen to reflect the way in which the algorithm proceeds iteratively by following an upward path from a low-valued pixel to a significant summit or peak in data-value. The following description of the algorithm uses this fell-walking metaphor at frequent intervals.

FellWalker is a form of *watershed* algorithm (Roerdink and Meijster, 2001)—a class of algorithms that segment images by identifying the "watershed" lines that separate low lying areas ("catchment basins"). FellWalker inverts this idea so that each identified segment of the array is associated with a peak (a local maximum), rather than a basin (a local minimum). It shares much in common with the HOP algorithm (Eisenstein and Hut, 1998)—another gradient-tracing watershed algorithm, but is designed for use with gridded observational data rather than particles in an N-body simulation. HOP is known to be relatively insensitive to the values supplied for its parameters (except the lower threshold), reflecting a similar feature found for FellWalker (see Section 3).

An implementation of the FellWalker algorithm is included in the Starlink CUPID[1] package (Berry et al., 2007; Berry, 2013, ascl:1311.007) together with implementations of other clump-finding algorithms such as GaussClumps (Stutzki and Guesten, 1990, ascl:1406.018) and CLUMPFIND. In common with the rest of the Starlink software (Currie et al., 2014a, ascl:1110.012) the source code for the CUPID package is open-source and is available on Github.[2] Pre-built binaries for the complete Starlink software

☆ This code is registered at the ASCL with the code entry ascl:1311.007.

E-mail address: d.berry@jach.hawaii.edu.

[1] http://starlink.jach.hawaii.edu/starlink/CUPID.

[2] https://github.com/Starlink.

http://dx.doi.org/10.1016/j.ascom.2014.11.004

Fig. 1. Wastwater and the Wasdale Fells, including Great Gable (centre–left) and snow-covered Scafell Pike, the highest point in England at 978 m above sea level, just visible under cloud.
©: Nick Thorne, http://www.lakedistrict.gov.uk/learning/freephotos#.

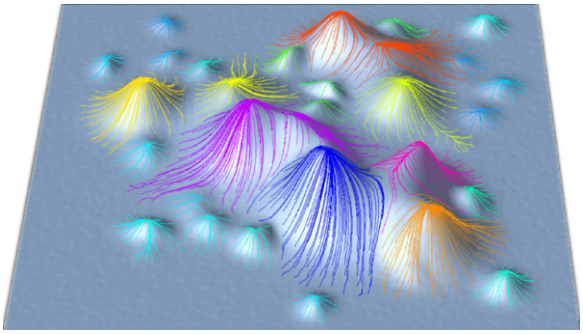

Fig. 2. In 2-dimensions, peaks in data value are often reminiscent of the fells of northern England such as those in Fig. 1. The FellWalker algorithm performs many walks starting at various low-land pixels, and for each one follows a line of steepest ascent until a significant summit is reached. All walks that terminate at the same peak are assigned to the same clump, indicated by different colours in the above figure. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

collection can be obtained from the Joint Astronomy Centre, Hawaii.[3] The native data format used by CUPID is the Starlink NDF (Jenness et al., in press), but FITS data can be handled transparently by means of the Starlink CONVERT package (Currie et al., 2014b).

FellWalker, like CLUMPFIND, segments the supplied data array into a number of disjoint regions, each associated with a single significant peak. Whilst this approach has been used widely, there are several alternative approaches to the problem of identifying clumps of emission which can be more appropriate, depending on the particular science being performed. For instance, it may be beneficial to allow clumps of emission to overlap (*e.g.* GaussClumps Stutzki and Guesten, 1990, ascl:1406.018 and GetSources Men'shchikov et al., 2012), or to take account of the hierarchical structure within clouds (*e.g.* dendrograms Rosolowsky et al., 2008). However, such issues are outside the remit of the FellWalker algorithm, and consequently this paper provides only a comparison of FellWalker with CLUMPFIND.

2. The FellWalker algorithm

The core of the FellWalker algorithm consists of following many different paths of steepest ascent in order to reach a significant summit, each of which is associated with a clump, as illustrated in Fig. 2. Every pixel with a data value above a user-specified threshold is used in turn as the start of a "walk". A walk consists of a series of steps, each of which takes the algorithm from the current pixel to an immediately neighbouring pixel of higher value, until a pixel is found which is higher than any of its immediate neighbours. When this happens, a search for a higher pixel is made over a larger neighbourhood. If such a pixel is found the walk jumps the gap and continues from this higher pixel. If no higher pixel is found it is assumed that a new summit has been reached—a new clump identifier is issued and all pixels visited on the walk are assigned to the new clump. If at any point a walk encounters a pixel which has already been assigned to a clump, then all pixels so far visited on the walk are assigned to that same clump and the walk terminates.

It is possible for this basic algorithm to fragment up-land plateau regions into lots of small clumps which are well separated spatially but have only minimal dips between them. The raw clumps identified by the above process can be merged to avoid such fragmentation, on the basis of a user-specified minimum dip between clumps.[4] These merged clumps may, optionally, be cleaned by smoothing their boundaries using a single step of a cellular automaton.

Finally, each clump is characterised using a number of statistics, and a catalogue of clumps statistics is created together with a pixel mask identifying the clump to which each pixel is assigned.

The following sections give more detailed descriptions of each of these phases in the FellWalker algorithm.

2.1. Identifying raw clumps

An array of integer values is first allocated, which is the same shape and size as the supplied data array. This "clump assignment array" (CAA) is used to record the integer identifier of the clump, if any, to which each pixel has been assigned. All clump identifiers are greater than zero. An initial pass is made through the supplied data array to identify pixels which have a data value above a user-specified threshold value. Such pixels are assigned a value of zero in the CAA indicating that the pixel is useable but has not yet been assigned to a clump, and all other pixels are assigned a value of -1 indicating that they are unusable and should never be assigned to a clump.

This initial CAA is then searched for any isolated individual pixels above the threshold. Such pixels are set to -1 in the CAA, indicating they should be ignored.

The main loop is then entered, which considers each pixel in turn as the potential start of a walk to a peak. Pixels which have a non-zero value in the CAA are skipped since they have either already been assigned to a clump (if the CAA value is positive) or have been flagged as unusable (if the CAA value is negative). A single walk consists of stepping from pixel to pixel until a pixel is reached which is already known to be part of a clump, or a significant isolated peak is encountered. The vector indices of the pixels visited along a walk are recorded in a temporary array so that they can be identified later.

At each step, the pixel values within a box of width three pixels are compared to the central pixel to find the neighbouring pixel which gives the highest gradient.[5] Thus 2 neighbours are checked if the data is 1-dimensional data, 8 are checked if the data is 2-dimensional and 26 are checked if the data is 3-dimensional. The gradient is evaluated in pixel coordinates, without regard to the physical units associated with each axis.

If the highest gradient found above is greater than zero – that is, if there is an upward route out of the current pixel – the walk steps to the selected neighbouring pixel. If this new pixel has already been assigned to a clump (*i.e.* if the CAA holds a positive value at

[3] http://starlink.jach.hawaii.edu/starlink.

[4] In common with other parameters, this minimum dip parameter is specified as a multiple of the noise level in the data.

[5] This gradient takes into account the fact that the centres of the corner pixels are further away from the box centre than are the centres of the mid-side pixels.