Solar Magnetic Tracking. I. Software Comparison and Recommended Practices

/DRAFT 1-Jul-2005 - hemisemifinal, for co-author review/

C.E. DeForest
Southwest Research Institute, 1050 Walnut Street Suite 400, Boulder, CO 80302
deforest@boulder.swri.edu,

H.J. Hagenaar
Lockheed Martin Advanced Technology Center, Org. ADBS., Bldg. 252, Palo Alto, CA 94304
hagenaar@lmsal.com

D.A. Lamb
Dept. of Astrophysical and Planetary Science, University of Colorado, Boulder 80309-0391 USA
derek@boulder.swri.edu

C.E. Parnell
School of Mathematics and Statistics, St. Andrews University, St. Andrews, Scotland KY16 9S
clare@mcs.st-and.ac.uk

and

B.T. Welsch
University of California at Berkeley Space Sciences Laboratory, 7 Gauss Way, UCB, CA 94720-7450, USA
welsch@ssl.berkekeley.edu

Abstract

Feature tracking and recognition are increasingly common tools for data analysis, but are typically implemented on an ad-hoc basis by individual research groups, limiting the usefulness of derived results when selection effects and algorithmic differences are not controlled. Specific results that are affected include the solar magnetic turnover time, the distribution of sizes and lifetimes of magnetic features, the physics of small scale flux emergence, and the small-scale dynamo. In this paper, we present the results of a detailed comparison between four tracking codes applied to a single set of data from SOHO/MDI, describe the interplay between desired tracking behavior and parametrization of vision algorithms, and make recommendations for selection and tracking practice in future work.
1. Introduction

The last decade has seen a sea change in the way that solar physics is accomplished. Advances in detector technology have permitted missions such as SOHO (e.g. Scherrer et al. 1995) and TRACE (Handy 1999), and ground-based observatories such as GONG (Leibacher 1995), to produce far more data than can be analyzed directly by humans. The planned SDO mission (Schwer et al. 2002) will produce data a thousand times faster. Hence, automated data mining has become a necessary tool of the analysis trade. Applied to image data, data mining consists of algorithmic recognition of visual features in the data. Applications such as feature and pattern recognition fall within the field of computer vision, which is the subject of active research in the computer science community.

Magnetic feature identification and tracking have proved useful for extracting statistical parameters of the solar dynamo (e.g. Hagenaar et al. 1999; Schrijver et al. 1997), allowing more sophisticated analyses than have been possible by hand (e.g. Harvey 1993). Current applications of feature tracking include characterization of bulk field behavior at the photosphere, probing of the solar dynamo, identification of the magnetic roots of solar atmospheric features, and constraint of MHD models. Each of these applications is discussed below.

In the last few years, each of our research groups has developed a separate computer code for magnetic feature tracking, optimized to our particular needs. Feature tracking is not a simple endeavor, and many subtle characteristics of a code can strongly affect the statistics of the results. Each of our codes was developed with some care with a particular aim in mind, but in an essentially ad-hoc manner; this has made it difficult to compare results across codes, or to duplicate scientific results across groups.

In November 2004 we met at St. Andrews University to reconcile results from each of our independent tracking codes: CURV (Hagenaar et al. 1999; Strous et al. 1996), MCAT (Parnell 2002), YAFTA (brian reference Welsh et al. 2004) and SWAMIS (Lamb and Deforest 2003). We applied each code to a sample data set and compared results from the different algorithms to reconcile the results across the codes. Furthermore, we identified how algorithmic choices affect magnetic feature tracking results, and developed a set of recommended practices to guide future development of feature tracking and related software for the solar community.

In this article, the first in a series on our tracking results, we discuss the state of the art and some current applications of magnetic tracking software. In §2 we outline the basic steps of a feature tracking algorithm; in §3 we describe our individual codes as they existed in November 2004; in §§4-5 we present and discuss the differences between the codes’ results as applied to a reference data set; and in §6 we recommend “best practices” for future codes to follow for typical feature tracking applications. Finally, §7 contains some general conclusions and insights, and a glossary at the end contains recommended vocabulary to describe specific aspects of magnetic tracking.
1.1. Current applications of magnetic feature tracking

Feature tracking is a fundamental component of many types of data analysis. Applied to the solar magnetic field, it has been used to characterize the statistical parameters of the solar magnetic field by determining the distribution of feature sizes and fluxes (Harvey 1993; Hagenaar et al. 1999; Hagenaar 2001; Parnell 2002) and the average lifetime of individual features (Hagenaar et al. 2003). Automated extraction of parameters such as clustering distributions (Lamb and Deforest 2003) and event distributions (DeForest and Lamb 2004) are being used to derive more detailed information about the solar dynamo. All of these applications are dominated by the relationship between small scale event detections and the noise floor of the instrument used for detection, generally a line-of-sight / scalar magnetograph such as SOHO/MDI (Scherrer et al. 1995) or GONG (Leibacher 1999).

Feature tracking is useful for constraining the energy input into flux systems in the solar corona. Much of the energy deposited into the chromosphere and corona is thought to be transported by the Poynting vector, as photospheric motions do work on the magnetic field by pushing magnetic flux around the surface (e.g. Parker 1988; Fossum and Carlsson 2004). Feature tracking allows simple derivation of the motion field from time series of images. Welsch and Longcope (2002) and Welsch et al. (2004) are applying feature tracking to the problem of modeling heating processes on the active region level, and DeForest and Lamb (2004) and Parnell (2000; 2002) are using feature tracking to identify the roots and nature of small scale heating events such as bright points.

Another application of feature tracking is to drive boundary conditions of semi-empirical MHD models of the solar atmosphere. Time-dependent MHD modeling requires knowledge not just of the three-dimensional vector field at the surface of the Sun, but also of the motion of individual lines of magnetic flux; feature tracking derives the motion information from time series measurements of the magnetic field. Indeed, Peano’s existence and completeness theorem (see, e.g., Simmons 1972) implies that knowledge of the initial magnetic topology in the force free upper layers of the atmosphere, together with the radial component of the field at the lower boundary, is equivalent to knowledge of the full vector field everywhere on the lower boundary. Provided that the initial topology may be estimated, this equivalence makes feature tracking a powerful tool for modeling energy input into the solar atmosphere even in the absence of full vector field measurements, as the 2-D distribution of magnetic flux on the photosphere approximates the distribution at the $\beta = 1$ surface in the chromosphere.

2. Discussion of Tracking Algorithms

Feature tracking can be divided into five separate operations: (i) image preprocessing; (ii) discrimination/detection; (iii) feature identification within a frame; (iv) feature association across frames; and (v) event detection. In addition, some noise filtering is accomplished by filtering the associated features to discard short-lived or small features that have too high a likelihood of being noise. Here, we discuss in a general way the components of any magnetic feature tracking algorithm.
2.1. Preprocessing

Reducing the noise floor and eliminating perspective effects requires preprocessing images before applying feature recognition. Temporal averaging, projection angle scaling, and resampling to remove perspective and solar rotation effects are commonly applied before most high-level analysis. In particular, 0.5-2 arcsecond scale magnetograms benefit from being averaged over 5-12 minutes to reduce background noise, and line-of-sight (Stokes V) magnetograms of the quiet sun benefit from being divided by a background factor to account for the difference between the magnetogram line of sight and local vertical at the surface of the Sun, under the model that weak field is close to vertical at the photosphere. Much existing work with digital solar images has used flawed resampling techniques that introduce artifacts into the data; a discussion on better resampling techniques may be found in DeForest (2004).

2.2. Discrimination

Any feature-recognition algorithm requires discrimination, i.e. the separation of foreground features from background noise. Every magnetogram sequence appears to contain many faint features at or slightly below the level of the noise floor, so discrimination is not trivial.

Most magnetograms made with a filtergraph type instrument such as MDI or GONG contain at least three sources of noise: (i) photon statistics, which produce a familiar white noise spectrum; (ii) P-mode contamination, which is due to the five-minute Doppler oscillations leaking into the Zeeman signal; and (iii) granulation noise, which is due to solar evolution between the different filtergraph exposures that make up each magnetogram. The photon shot noise is a uniform blanket random variable with an independent sample at every pixel and a presumed Gaussian distribution. The P-mode contamination is a random variable with far fewer independent spatial samples per image, because of the low spatial frequencies of the P-modes, and an oscillating temporal component. Granulation based noise has a spatial scale of a few arc seconds and a coherence time of 5 minutes. MDI is well tuned so that the three sources of noise are about equal in individual images; but in spatially binned, temporally averaged, or smoothed images, the granulation and P-modes dominate the noise spectrum.

The simplest discrimination scheme, direct thresholding, works well only for strong magnetic features that are well separated from the noise floor, such as network flux concentrations. Other types of magnetic feature, such as weak intranetwork fields, suffer because keeping the threshold high enough to avoid false-positive detections creates a large number of false-negative non-detections of the weak magnetic features. The problem is the huge number of individual detection operations (one per pixel per time step), which makes false positives a significant problem. With a Gaussian noise distribution, setting the threshold to three standard deviation (σ units yields a false positive rate of about 10^{-4}, so that a 300-frame data set with dimension 300x300 pixels would yield around three thousand false positive detections from noise alone, and perhaps 10-30 times that number of inconsistently detected weak features (false negatives).

We have adopted two different schemes to work closer to the noise floor, both of which add additional
tests to the basic threshold test. The first is the curvature method used by Strous et al. (1996) and by Hagenaar et al. (1999), in which both the data values and their second derivative are tested. To be considered part of a local maximum/minimum by the curvature algorithm, the magnitude of a pixel must exceed a value threshold and all surrounding pixels must have a negative/positive second derivative in each of the horizontal, vertical, and two diagonal directions. The second derivative criterion adds four additional independent threshold tests for each pixel, reducing the number of false detections at a given threshold and allowing lower threshold values.

The second method we have adopted is hysteresis, used by Lamb and Deforest (2003) and by Parnell (2002), in which two thresholds are applied: a high threshold for isolated pixels, and a second, lower threshold for pixels that are adjacent to already-selected pixels. Adjacency is allowed in space and/or time. The hysteresis method misses weak features, but captures every feature that at some point (depending on the criteria imposed) exceeds the large threshold. The proximity requirement reduces the number of pixels that undergo the lower threshold test, and therefore reduces the number of false-positive detections. Depending on application, the higher threshold is chosen to be \(3 - 6\sigma\), and the lower threshold \(1 - 3\sigma\), where \(\sigma\) is the RMS variance of the data.

2.3. Feature identification

Feature identification is the operation of connecting masked pixels into distinct identifiable (and identified) structures in each frame. In practice, this means forming an image whose pixels have integer numeric values that correspond to index numbers of particular features. Each of our codes uses a variant of a clumping algorithm that identifies connected loci of pixels within a masked region. MCAT clumps masked pixels directly into contiguous regions. YAFTA and SWAMIS can switch between direct clumping and a gradient based (“downhill”) method that dilates local maxima by expansion down the gradient toward zero flux density. CURV also uses direct clumping, but generates initial feature masks with a data-value curvature method that restricts the features to single mounds, yielding features that are segmented more like those of the downhill method in YAFTA and SWAMIS than like the other clumping codes. All three techniques are illustrated in Figure 1.

The tradeoff between the clumping and downhill methods is that the downhill method is better at picking out the structure of individual clusters of magnetic flux, while the clumping method is somewhat less noise-susceptible. Fluctuations from either solar convection or instrument noise can easily create small local maxima that are identified as transient structures by the downhill method; simple clumping eliminates these small transients, for better or for worse. Which method is appropriate depends on the specific scientific application, as discussed further in Section 6.
2.4. Feature association

Feature association is the fixing of a feature’s identity across different frames of an image sequence. Most features in adjacent frames of an image are related by similarity: when a feature in frame $m + 1$ is sufficiently similar to the feature in frame $m$, then it is likely that the two features represent the same physical object at the two different times. All of our existing codes use variations of a dual-maximum-overlap criterion to identify persistent features across frames (Figure 2); this technique associates two features $B$ and $C$ in adjacent frames only if $B \cap C$ is larger (in a flux-weighted sense) than any other intersection with either $B$ or $C$.

It is notable that no local overlap algorithm agrees with a human observer in all cases, as human observers use more information than strict overlap – including something like a predictor/corrector position algorithm. Maximum overlap works well in the case where the motion of all features is small compared to their width divided by the time step. If the time step is too long, small features can move more than their diameter in a single frame, leading to the mistaken identity problem, where a single visually identifiable feature frequently changes its identity in the tracked code. In such cases, one can (i) use faster frame rates, (ii) generate dilated feature masks for association, (iii) use linear location extrapolation to account for the large interframe motion, and/or (iv) use a minimum-distance criterion rather than maximum overlap.

Clumped features can overlap in quite complex ways; some of our codes use special handling for these complex features, which is described in more detail in the description of each code.

2.5. Filtering based on size/longevity

When working close to the noise floor, it is useful to reject small features, because false positives are much more likely in small clusters of pixels than in large ones. All of our codes reject identified features that do not meet some minimum size criterion. Criteria that are useful include: maximum size; average size; lifetime; or total number of pixels across the life of the feature. The filtering can be accomplished

![Fig. 1.— Effect of different feature-identification schemes on the identified structure of a large flux concentration. (A) Clumping identifies all connected above-threshold pixels into a single feature. (B) Downhill methods identify one feature per local maximum region. (C) Curvature methods identify the convex core around each local maximum.](image-url)
Fig. 2.— A pathological association case. Features A and B are in the previous frame, C and D in the current frame. A maximum-overlap method associates B and C. The recommended associative algorithm (dual-maximum overlap) associates B=C if and only if $B \cap C$ is the largest of C’s intersecting regions and also the largest of B’s intersecting regions. A and B merge to form C, at the same time that D calves via fragmentation from B.
only after feature identification (for per-frame size checks) or feature association (for maximum size checks or longevity checks). When rejecting an identified feature, one must be careful to avoid the Swiss cheese problem, which results when fluctuations in the interior of a large structure result in identification of strong but short-lived features that are then rejected for having short lifetimes. Such rejections leave undetected holes in the larger structure. The Swiss cheese problem may be overcome by re-associating a short-lived feature with a surrounding larger feature if there is enough overlap (e.g. SWAMIS); or by time-averaging the data so that the frames oversample the time axis, reducing the frame-to-frame fluctuations of individual features (CURV). (need figure)

2.6. Classification of Origin and Demise

Identifying and tracking individual features is properly described as “mere” feature tracking, but identifying individual types of event that may comprise several features is more properly described as a complete computer vision application. Not all of our feature tracking codes include provision to detect and identify interactions of multiple flux concentrations, such as pairwise emergence, but such detection is an important part of characterizing magnetic evolution and hence is discussed here. In particular, because magnetic features are not corks but rather cross-sections of curvilinear manifolds (field lines that pass through the photosphere), they are connected pairwise by the magnetic field. Identifying the association between freshly emerged pairs thus gives useful information about the overall field topology and how it changes via reconnection of the overlying field before the death (e.g. by submergence) of the individual features.

The photospheric magnetic field appears to be due to flux tubes that emerge from below the surface of the Sun (Harvey and Martin 1973; Harvey 1993; Chae et al. 2001); such events give rise to two oppositely signed magnetic features that grow together and separate in a divergent surface flow (Hagenaar 2001; Hagenaar et al. 2003; Simon et al. 2001), and the origin detection code in SWAMIS and in CURV was originally intended to identify such events. However, not all magnetic features are observed to originate with this balanced emergence mechanism. New small features can also form by fragmentation of pre-existing large features into like-signed fragments; this process is also called calving if the new feature is small compared to the surviving feature. Furthermore, many features simply appear, without any surrounding flux at all or in ways that appear to violate flux conservation. The nature of these appearances will be considered in detail in a later paper in this series.

The origin and demise of features is different than the origin and demise of magnetic flux itself: in particular, features can fragment or merge under the influence of the photospheric flow field (e.g. Schrijver et al. (1997)) without any flux emerging or submerging through the photospheric surface.

Software to identify origin events recognizes flux concentrations near each newly detected concentration, and classifies the origin according to these nearby concentrations and their flux time derivative. To avoid missing associated structure, an allowed margin of error is required in the spatial or temporal offset between two associated features, and also in the flux rate-of-change between the features.

Demise events are similar to origin events and may be recognized with the same code, operating on
tracked data in reverse time order. As with birth events, demise events are not necessarily related to emergence or submergence of magnetic flux.

3. Description of Codes

We have independently developed four separate feature-tracking codes. The codes were developed for slightly different goals and hence each has a slightly different approach and yields different results when analyzing the same data. Before reconciling the differences, we devote a brief subsection to each of these individual codes.

3.1. CURV: Curvature Recognized Vertices

CURV is a suite of IDL procedures developed by Strous et al. 1996 and Hagenaar (2001) for analysis of the magnetic network and turnover of the small-scale field. It uses a curvature based discriminator and a pixel-clumping technique to identify objects. The combination yields a hybrid of the downhill and clumping mask techniques used by the other codes, with some of the advantages of each for small scale magnetic features. CURV is optimized for small, weak magnetic features: the curvature discrimination results in well-separated features that are readily associated by pixel clumping, with the tradeoff that clusters of features are difficult to identify because they are separated by a non-convex margin.

The data are treated as IDL data cubes. The output is a cube containing per-image unique feature labels, and a tabulated list of features found in each frame with some ancillary data suitable for post processing and lifetime analysis.

3.1.1. CURV discrimination

The curvature $\mathcal{H}$ of the intensity $B$ at pixel coordinates $(i, j)$ in a certain direction $(h_1, h_2)$ is calculated as the negative spatial derivative, given by

$$\mathcal{H}(h_1, h_2) = 2B(i,j) - B(i+h_1,j+h_2) - B(i-h_1,j-h_2),$$

(1)

where $(h_1, h_2) = (1,0), (1,1), (0,1), (-1,1)$. Around a local maximum, $\mathcal{H}$ is positive in all directions, for a minimum it is negative. A coherent patch where $\mathcal{H}$ is positive everywhere or negative everywhere is considered as a possible concentration. It is selected as a concentration when $|B|_{max} B_{thr}$, where $|B|_{max}$ is the strength of the strongest pixel in the patch and $B_{thr}$ is a threshold flux density.

CURV finds only the positive-curvature cores of magnetic features, and not the whole feature. The total flux of each feature is derived by multiplying the core flux value by a factor of 3. This scaling factor was determined based on a Gaussian shape model for small magnetic features as described in Hagenaar et al. (1999).
3.1.2. **CURV feature identification**

CURV identifies features as connected collections of pixels that meet the curvature discrimination criterion. Coordinates, area, and flux content are determined based on flux “center of gravity” measurement in each feature. The area is measured in pixels. Only concentrations with a size of at least 9 pixels are selected. A perfectly round concentration with an area of 9 pixels would have a diameter of 3.4 pixels. This step yields a set of detected feature maps: images whose pixels have a value of either 0 or a feature number that is unique within each image in the sequence.

3.1.3. **CURV feature association/tabulation**

CURV uses a tabular system to associate features between frames, and simultaneously identify formation and demise events. For each detected-feature map $m$ with concentrations labelled $n_i$, the algorithm identifies corresponding concentrations labelled $n'_j$ in map $m + 1$. The labels of concentrations $n'_j$ are changed to the same values as the labels of the overlapping concentrations $n_i$. The labels are registered in a large, initially empty matrix. Every time step, the next row is filled.

**Initialization:** The first label map $m = 0$ in the sequence defines the initial values of the tracking matrix and of the label maps. For example, if the first map contains concentrations $1 \ldots N$, the tracking matrix is initialized to

$$
\begin{pmatrix}
1 & 2 & \ldots & N \\
0 & 0 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots 
\end{pmatrix}
$$

(2)

**One-to-one correspondence:** The algorithm identifies all concentrations $n'_j$ in map $m + 1$ overlapping any part of concentration $n_i$ in map $m$. In the simplest case, only one concentration $n'_j$ overlaps only one concentration $n_i$. Concentration $n'_j$ is given label $n_i$ and the corresponding element of row $m + 1$ of the matrix is $n_i$. In Table 1, these are the concentrations with labels $n_1$.

**New concentration:** New concentrations in map $m + 1$ have no overlap with any concentration in map $m$. They are given a new label $n_k$, and a new column in the tracking matrix. The $0 \ldots m$ rows of these concentrations stay 0 (labels $N + 2$ in Table 1).

**Disappearing concentration:** The concentrations $n_k$ in map $m$ without overlap with concentrations in map $m + 1$, have disappeared. The rows $m + 1 \ldots$ and higher stay 0.

An example in Table 1 is the label $n_3$, disappearing in row $m + 1$.

**Fragmentation:** If two or more concentrations $n'_j, n'_j, \ldots$ overlap concentration $n_i$, concentration $n_i$ has fragmented. One of the resulting concentrations, $n'_{j_1}$, is given label $n_i$, the others are assigned new labels, $n_{k_1}, n_{k_2}, \ldots$. For all $n'_{j_2}, \ldots$ a column is appended in the matrix.
<table>
<thead>
<tr>
<th>row #</th>
<th>1-to-1 fragment</th>
<th>disappear</th>
<th>collision</th>
<th>collision</th>
<th>fragment</th>
<th>new</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m-1$</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$m$</td>
<td>$n_1$</td>
<td>$n_2$</td>
<td>$n_3$</td>
<td>$n_4$</td>
<td>$n_5$</td>
<td>...</td>
</tr>
<tr>
<td>$m+1$</td>
<td>$n_1$</td>
<td>$n_2$</td>
<td>$0$</td>
<td>$n_4$</td>
<td>$n_4$</td>
<td>...</td>
</tr>
<tr>
<td>$m+2$</td>
<td>$n_1$</td>
<td>$n_2$</td>
<td>$0$</td>
<td>$n_4$</td>
<td>$0$</td>
<td>...</td>
</tr>
<tr>
<td>$m+3$</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>$0$</td>
</tr>
</tbody>
</table>

Table 1: CURV association matrix containing the labels $n_i$ of all tracked concentrations (across column), as a function of time $m$. Equivalent matrices are determined containing the flux, area, minimum signal, maximum signal, and the $(x, y)$ coordinates of the corresponding concentrations. Column 1 shows a persistent, surving flux concentration; the other columns show examples of each type of interaction. In frame $m$, the concentration labeled $n_2$ fragments into two bits, one of which is given the label $N + 1$; concentration $n_3$ disappears; concentrations $n_4$ and $n_5$ collide; and concentration $N + 2$ appears.

All fragmentation products are identified with the initial concentration $n_i$. Rows $0 \ldots m - 1$ of these concentrations are 0, and their $m^{th}$ rows copy row $m$ of concentration $n_i$. The $m + 1$ rows get the values $n_{k_1}, n_{k_2}, \ldots$.

For example, in Table 1 concentration $n_2$ fragments to form $n_2, N + 1$.

Colliding concentrations: The fragmentation analysis is applied in reversed time direction, as shown in Table 1 for concentrations $n_4$ and $n_5$. The column associated with concentrations $n_4$ continues with the label $n_4$. The other, $n_5$ is marked once by $n_4$, in order to distinguish disappearing concentrations from colliding and merging concentrations. From there, it stays 0.

Three-particle interactions: CURV does not consider three-particle and higher interactions.

The properties of the concentrations are also put in matrices: two coordinate matrices, a flux matrix, and an area matrix.

### 3.2. MCAT: Magnetic Clumping Associative Tracker

MCAT is a suite of routines in IDL, that were developed by Parnell to study quiet-sun magnetic fields. MCAT makes several passes through the data to accomplish each of the main steps described in §2. In typical use, the data are preprocessed by despiking, time averaging, and spatial smoothing; this preprocessing step was omitted for the current study, which focused on the response of each tracking code to a single prepared data set.
MCAT uses a two-threshold signed discriminator with hysteresis in space and a clustering restriction. Pixels are subjected to the high threshold unless they are adjacent to the detected locus in the previous frame, in which case they are subjected to the low threshold. The thresholds are specified in $\sigma$ units of the background noise distributions. Only flagged pixels with more than a specified number of flagged neighbors are retained as part of the discrimination mask of each frame.

3.2.1. MCAT Discrimination

MCAT first constructs a histogram of the pixel values for the entire data set. A Gaussian distribution is fitted to the core of this histogram to model the noise distribution in the data. The half-width at half-maximum (HWHM) is taken to be the $1\sigma$ noise level in the data, and the discrimination thresholds are specified in these $\sigma$ units as $n_h\sigma$ and $n_l\sigma$. Generally, $n_h = 4$ and $n_l = 2$; however, these values are adjustable to suit the particular analysis to be undertaken.

The discrimination itself takes place by a single forward and a single backward pass through the data. Each pass considers the frames in forward or reverse order. In each frame, a raw detection mask is assembled by finding all pixels that are greater than $n_h\sigma$ or that are greater than $n_l\sigma$ and also intersect a 1-pixel dilation of the mask found in the previous frame (for frames other than the first one in the pass). At the end of consideration of each frame, the current mask is filtered: only pixels with more than a specified number of neighbors $n_p$ above $n_h\sigma$ are kept at each step. In typical use, $n_p = 3$ out of 8 possible neighbors. This has the effect of rejecting very small features, reducing the false positive rate. Separate masks are collected for positive and negative features in the data.

3.2.2. MCAT Feature Identification

The MCAT discriminator makes a single forward pass and a single reverse pass through the data. These are combined into single masks for each polarity using the logical OR of the forward mask and the reverse mask of that polarity. (***) This yields a data cube in which the pixels of an identified feature are labelled with a unique integer (whose sign is the same as the polarity of the feature). All other pixels have a value of zero.

Clare - still need input on how the feature identification works? As of the (***) above, it sounds like we have a data set filled with 1, 0, or -1 depending on whether there is a positive feature, no feature, or a negative feature there is as yet no explanation of how the ID integers are assigned.

3.2.3. MCAT Feature Association

MCAT associates individual features across frames using a maximum overlap method. Two passes are made through the data. The first examines the loci from the feature identification step, and the second is a
checking run to prevent the multiple fragmentation/coalescence of features.

In the first pass, the code loops over all features in each frame after the first frame, and for each one performs the following three steps:

(i) If two features have a one-to-one overlap then the feature in frame \(m\) is assigned the same numeric label as the corresponding feature in frame \(m - 1\).

(ii) If there are two or more features in frame \(m\) overlapping just one feature in frame \(m - 1\), then the one with the greatest flux-weighted overlap with the \(m - 1\) feature takes on the label of the previous feature. The second feature is assigned a new unique label and is said to have originated through fragmentation. A separate table contains a record of the frame in which the fragmentation occurs, and which features the fragmentation involves.

(iii) If there are two or more features in frame \(m - 1\) overlapping just one in \(m\), then the feature in \(m\) takes on the label of the feature with the greatest flux-weighted overlap in \(m - 1\). The second fragment is said to have died through merging. A separate table contains a record of the frame in which the merger occurs and which features the merger involves. In the complex situation depicted in Figure 2, Feature C in frame \(m\) would take on the label of B, feature A would be said to have died, and feature D would be said to have originated through fragmentation. In more complex situations, the dual-maximum-overlap criterion is used for primary association and all other features are said to either coalesce or fragment as appropriate.

In the second pass, several types of anomaly are detected and removed through “cleanup” passes through the data, in a further three steps per feature.

(i) Features may have completely-surrounded holes in the center of the feature. If such a hole is found then the missing pixels and their fluxes are labelled and counted as part of this feature provided that they have the same polarity as the feature. This is assumed to have occurred through fluctuations in the signal lowering the pixel value below the lower detection threshold in the presence of a real feature.

(ii) Situations may occur in which two features merge, then split again, and may oscillate between these two states over several frames. To reduce the occurrence of these misleading birth and death events, features are prevented from merging if the merger occurs just before a split. In such cases, the identity of the feature that would originally have died through merging is retained and is propagated by relabelling overlapping pixels in successive frames.

(iii) Situations may also occur where a feature first splits and then merges back into a single feature. In these situations, the two features are assumed not to have actually split and are glued together by picking up extra pixels of the correct polarity.

The last two steps, in particular, need to be performed over the whole dataset in the order given, to maintain stability of the feature tracking.
3.2.4. MCAT tabulation

MCAT makes a final pass through the data to record the characteristics of each feature in each frame in a series of arrays that contain the feature labels, areas, fluxes, locations (flux weighted center of gravity and simple midpoint), and peak field strengths for each feature in the frame. To minimize wastage of memory, the arrays are packed with a single layer of address indirection. Each array is $M_f \times N_{\text{max}}$ (row, column order), where $M_f$ is the number of frames and $N_{\text{max}}$ is the maximum number of features in a single frame throughout the data set. An $M_f \times N_{\text{max}}$ label array contains the unique label of each feature from a particular frame (padded with zeroes if $N < N_{\text{max}}$) and is used as a row-wise index into the columns of the array. Compared to simple row/column addressing, this results in a memory savings of $N_{\text{max}} / N_{\text{total}}$, where $N_{\text{total}}$ is the number of features identified in the entire dataset. Individual timeseries are retrieved by single indirection, for example using the “wherem” command in IDL.

Further tables are maintain identifying features born or destroyed through fragmentation and coalescence. Finally, the locus of each feature is stored as a collection of pixel indices in a separate array to facilitate coloring the features for visualization.

3.3. SWAMIS: the Southwest Automated Magnetic Imaging Suite

The Southwest Automated Magnetic Imaging Suite (SWAMIS) is a free package of Perl/PDL modules that was developed by DeForest and Lamb as a tool to identify clustering and behavior of small-scale field elements. It makes five passes through the data: detection, identification, association, tabulation, and event classification.

The data are treated as Perl lists of images, which may be present in memory or handled as PDL::DiskCache objects that are swapped out to disk when not in use. Each pass refers to only a small “peephole” of 1-3 frames at a time, with limited backtracking, so that SWAMIS may be applied to arbitrarily long time series. The output is a collection of feature-label images with globally unique labels, a table listing the history of each feature across frames, and a separate table summarizing the origin and demise of each feature.

3.3.1. SWAMIS Discrimination

Like MCAT, SWAMIS uses a two-threshold signed discriminator with hysteresis in space and time. The SWAMIS hysteresis is somewhat more lax: pixels are subjected to the high threshold unless they are adjacent to any already-detected pixel, in which case they are subjected to the low threshold. The discrimination is iterated, dilating detected regions by comparison of their boundaries to the lower threshold, until no more points are added to the detected locus.

1The code may be downloaded from the URL “http://www.boulder.swri.edu/swamis”; or contact DeForest.
The high threshold is typically set to $6\sigma$ of the measured noise, and the low threshold is typically $2-3\sigma$. The discriminator keeps track of sign, yielding masks that are either 0, 1, or -1 depending on whether each feature is positive or negative. The threshold is specified to the code in absolute data value units: noise characteristics are determined in advance.

The neighborhood of a pixel comprises up to 26 pixels in a $3 \times 3 \times 3$ (x, y, t) cube, and SWAMIS can be switched to include different sorts of neighbors. Typically one does not consider the eight pixels that comprise the corners of the cube and that only share a single corner with the central pixel. The low threshold applies only to neighbors with the same sign as the adjacent region. The code can be made consider neighbors only within the same frame as the test pixel, preventing timewise hysteresis.

SWAMIS uses recursion to extend features as far as possible into an output detection mask. First, all pixels above the high threshold are marked in the output mask, then multiple scans are made of each image. At each iteration, all pixels adjacent to newly-detected pixels are tested against the low threshold. Because neighbors are allowed on the time axis as well as the spatial axes, a recursive algorithm is used to step back and forth through the data as new points are detected (requiring points in the previous frame to be checked against the lower threshold). There is a user-settable limit on how far back this “look-behind” can traverse; it is typically set to 20 frames.

3.3.2. SWAMIS Feature Identification

SWAMIS contains two identification methods: “clump” and “downhill”. The clump method groups any adjacent same-sign pixels into one feature with one identification number. The downhill method finds local maxima in the absolute data and assigns each maximum a unique label number. Label value then flows from each maximum toward zero until it either reaches the end of the feature or reaches a “valley”. In the event that two or more identification numbers meet in a valley, any pixel that could go to either is arbitrarily assigned to the lower of the two identification numbers.

The clump identifier also begins by seeding each local maximum with a unique label, but allows label value to flow “uphill” as well as down, so that contiguous blocks of detected pixels all end up carrying the lowest numerical label assigned to any of the local maxima within each block. This step produces a list of images, each of which contains features marked by integer labels that are unique with the frame. Positive features get positive labels; negative features get negative labels.

3.3.3. SWAMIS Feature Association

SWAMIS scans through the identification frames and identifies intersections of same-sign features between each frame $m$ and the successive frame $m + 1$. For each feature $f$ in frame $m$, a “forward intersection list” of intersecting features in frame $m + 1$ is assembled. For the largest of those intersecting features, $f'_{\text{max}}$, a “reverse intersection list” is assembled of features in $m$. $f'_{\text{max}}$ is assigned the same label as $f$, if
and only if the largest reverse intersector of $f'_{\text{max}}$ is $f$ (see Figure 2). After all such features have been found and associated, the remaining unassociated features are presumed to be new and are assigned new globally unique labels. The output of this step is a set of images marked with globally unique labels that are persistent across frames.

3.3.4. **SWAMIS Feature Tabulation**

SWAMIS separates the acts of associating features and tabulating feature histories into two separate passes. In the tabulation pass, SWAMIS calculates the flux, size, and center-of-flux of each feature for each frame. Features with lifetimes less than 4 frames, maximum sizes less than 4 pixels, or x,y,t volumes less than 8 pixels are discarded. To avoid the “swiss cheese problem”, such discarded features are re-associated: if a discarded feature is touching another non-discarded feature of the same sign then all of its pixels are reassigned to the non-discarded feature. The output of this step is a table with feature ID running across the top and time step running vertically, containing, the location and size information for the feature for each frame. The associated-label images can be touched-up in place to save disk space.

3.3.5. **SWAMIS History summarization**

SWAMIS’ final pass through the data is an attempt to extract histories for all features. We determine how each feature is born and dies. Possible birth methods are emergence, appearance, fragmentation, complex, and error. An emergence is said to occur if there is nearby opposing flux that also emerged within an adjustable window of time, or a nearby feature of opposite sign that increases in flux. An appearance is an origin with no nearby flux of either sign. A fragmentation is an origin associated with a nearby (or reverse-intersecting) feature of the same sign that decreases in flux. A complex birth is one for which there are several nearby concentrations, so that the code has difficulty sorting out which of the nearby features is associated with the newly-born one. An erroneous birth is one for which there are nearby features, but flux is not conserved.

Conservation of flux is enforced only approximately, to allow for noise and the difficulty of calibrating total flux. The allowed mismatch in flux is settable by the user for each of these types of event.

Feature death is categorized as submergence (or cancellation), disappearance, merging, complex, and error, respectively; the criteria are exactly the same as the birth criteria, but run backwards in time.

3.4. **YAFTA: Yet Another Feature Tracking Algorithm**

YAFTA was developed by Welsch to study motion of large-scale magnetic features near active regions; but it is also applicable to the network and quiet sun. In version 1.0 of the YAFTA code, feature identification and association are separate tasks; features are discriminated with a simple threshold; and a ranked downhill
aggregation (RDA) method is used to identify features.

### 3.4.1. YAFTA discrimination

Features are discriminated using a simple threshold to generate independent positive and negative feature masks. Future versions may use hysteresis. Negative and positive features are identified using separate images to which a threshold mask has been applied — pixels below threshold in absolute data value are set to zero. In the negative-feature case, the absolute value of the masked data is taken before thresholding is applied and the feature identification algorithm is called. Once masked, a two-dimensional label array is created, identical in size to the data slice.

### 3.4.2. YAFTA Feature Identification

YAFTA 1.0 uses a gradient-based, ranked downhill aggregation (RDA) method, introduced by Welsch and Longcope (2002), to group pixels into “hills” and “valleys,” in a two-dimensional data slice. Though its grouping mechanism is distinct, this downhill method is similar to watershed methods. Future plans include a clumping mechanism for feature identification.

Pixels in the masked data array are ranked by data value. Proceeding in ranked order, the data value of each pixel and its 8 nearest neighbors are then compared. If a pixel has no higher-valued neighbors, it is a local maximum, and a new label is created in the corresponding pixel in the label array. YAFTA rejects weak features by blocking new-label creation for local maxima that are lower than a high feature threshold.

If a discriminated pixel is not a local maximum, then it takes the label of the neighbor with the highest data value — which is guaranteed to have been assigned, by virtue of the rank ordering. The search proceeds until all pixels above threshold have been checked.

In a clumping scheme, a check would be implemented here: if a given pixel has at least two unique, previously-labeled neighbors, it would take lowest neighboring label as its own, as would all pixels bearing the other label or labels.

In this way, in a single pass, all features can be identified — using either a single or dual threshold, and either a downhill method or clumping.

### 3.4.3. YAFTA Feature Association

YAFTA associates features across features using the amount of spatial overlap between features in adjacent frames. For each feature in the data slice at \( m + 1 \), a binary mask is created with the feature’s pixels set to one, and this mask is multiplied by the mask of signed labels from the data slice at time \( m \). A feature at step \( m + 1 \) is then given the label of the same-signed feature from step \( m \) with which it has the
most spatial overlap; if it does not overlap any feature from step $m$, it is given a new label.

In cases where feature displacement might not allow overlap of features that should be associated, a dilated binary mask can easily be formed by by smoothing the initial binary mask and rounding up the resulting array with a ceiling function.

In the case where a feature at $m + 1$ intersects multiple features at $m$, assignment is to the lowest-labelled feature — an essentially arbitrary choice. If multiple features at $m + 1$ intersect a single feature at $m$, they are all initially assigned the same label.

After all features in $m + 1$ are considered, the new labels are scanned for duplicates — an indication of fragmentation. When multiple labels are present, the largest fragment (in area) keeps the disputed label, and the remaining fragments are assigned new labels.

Next, a search is done for labels from step $m$ that do not appear in labels from step $m + 1$. If a non-appearance is found, a reverse overlap is calculated. If there is no overlap, it is assumed the feature has disappeared. If there is at least one overlap, the source features of the overlapping features from $m + 1$ are searched, in order of decreasing overlap, for the first recorded judgement of fragmentation. The first such judgement of fragmentation, if any, is deemed a mistake, and the feature from $m + 1$ takes the label of the formerly-unpropagated feature from $m$. If no recorded fragmentations were found among the overlapping features from $m + 1$, a merger is assumed to have occurred.

YAFTA 1.0 stores origin, propagation, and demise information in two signed integers per feature per time step, one in a “source” field and one in a “termination” field. If a feature is matched one-to-one, or is the largest fragment of a fragmentation, its label is its source. If a feature is formed by fragmentation, but is not the largest fragment, the label of the largest fragment is stored as its source. If a feature appears at step $m$, then $-m$ is its source. Analogous entries for one-to-one matchings, mergings, and disappearances are stored in the termination field.

4. Tracking Results: A Comparison Across Codes

4.1. Description of the Dataset

We analyzed a sequence of 600 one-minute-cadence MDI high resolution quiet-sun images from 2003 June 04, beginning at 05:43 UT. The images were resampled into heliographic longitude/latitude coordinates (plate caree projection) using an orthographic model of the solar image, as shown in Figure 3. The reprojection used the ANA language resampling tools by Shine (1999). The tracked images were 300x300 pixels and ran over the range $-16.3^\circ$ — $-7.3^\circ$ in longitude and $-2.8^\circ$—$6.2^\circ$ in latitude. This scale slightly enlarged the images, to a pixel size of 0.03 heliocentric degrees (about 0.48 observer arc seconds at disk center). Images were derotated to the central time in the data sequence, using the Snodgrass (1983) synod differential rotation curve and rigid-body rotation at a latitude of $14^\circ$. These derotated, plate caree images were averaged together in blocks of five minutes each to reduce shot noise and P-mode interference.
The noise level was determined by fitting a Gaussian profile to the weak portion of the pixel strength distribution curve (Figure 4). The width ($\sigma$) of the best fit Gaussian profile was 18.3 Gauss, which should be taken as the sum of all incoherent noise components (principally shot noise, granulation, and P-mode leakage). Small frame-to-frame offsets of the zero point (presumably due to variations in the instrument’s exposure time) were found by measuring the offset from zero for the best-fit Gaussian, and removed by subtraction from each frame.

### 4.2. Network Characteristics

The simplest comparison to make across codes is distribution of sizes of detected magnetic features. Figure 5 shows the results of applying all four of our codes (with two different identification techniques for SWAMIS) to the same data. The five different techniques yield obviously different size distributions for the network; here we discuss the features in the plots and the differences between them. The plots all have the same height scale and the same bin size, so the histograms are directly comparable. All the codes exhibit high and low threshold behaviors that are discussed below; but it should be immediately apparent by inspection of Figure 5 that the codes diverge at the small end of the size spectrum, achieving a moderately good agreement in slope only for flux concentrations larger than about $2 \times 10^{18}$ Mx. All four methods produce a slope of about $-0.35 \pm 0.05$ decades per $10^{18}$ Mx, corresponding to an e-folding width of about $1.2 \times 10^{18}$ in the distribution.

All of the codes display a weak-feature threshold effect (false negatives) as small features that are close to the noise floor are eliminated by the discriminators. YAFTA and MCAT show the strongest threshold effects, because they rely on a combination minimum strength and minimum feature size in each frame to eliminate false-positives from the discrimination step. The YAFTA threshold is particularly abrupt because the initial detection discriminator uses no hysteresis, so that all detected features must have a minimum number of pixels with a minimum amount of flux per pixel. MCAT’s threshold is softer because the hysteresis feature of the discriminator allows weaker pixels to be detected around a strong core. CURV shows a still softer turnover and threshold because the CURV discriminator allows much weaker features to be detected: while the maximum-strength pixel in each feature must exceed a detection threshold, there is no strength threshold on the pixels surrounding the local maximum at the center of the feature. SWAMIS shows no obvious threshold at all because its recursive temporal hysteresis admits many features that have no strong pixels in a particular frame: provided that a feature has a single strong pixel at any point in its lifetime, all of its pixels are subjected to the weaker threshold.

The disagreement between the different codes on the weak feature distribution is telling: it is difficult to distinguish reliably the size distribution of magnetic features that are smaller than about $10^{18}$ Mx in strength even with time-averaged and conditioned MDI data. The MDI hi-res 1 $\sigma$ detection threshold in our time-averaged data is about $2.2 \times 10^{16}$ Mx, corresponding to a single pixel with an 18 Gauss signal (Figure 4); features with less than 50$\times$ this much flux are not reliably detected across methods.

The greater weak-feature counts of CURV and YAFTA compared with SWAMIS and MCAT do not
Fig. 3.— The tracked field of view, in context. Note that the edge of the grey circle in the MDI full-disk image is not the limb of the Sun, it is a crop radius for the instrument, just outside the limb.

necessarily correspond to greater sensitivity: our data set was not controlled for false positives. When characterizing a code for weak feature sensitivity, one should use noise injection null techniques to identify false positive rates. Likewise, despite the lack of obvious threshold the SWAMIS weak-feature distribution curve should not be trusted below about $10^{18}$ Mx because the hysteresis requirement may reject many
Fig. 4.— Distribution function of weak-field pixels in the test dataset, a sequence of five-minute average high resolution magnetograms from SOHO/MDI. A Gaussian fit to the low-valued pixels (presumed to be noise) is shown. The measured standard deviation (\( \sigma \)) of the images is 18.3 Gauss.

Fig. 5.— Network feature size distributions as derived by the five algorithms we compared in §3. See text for full discussion.

Transient weak features that never happen to achieve the flux density required to trip the high threshold.

The large-feature performance of the codes varies somewhat across algorithm, although the differences are not as gross as in the small-feature regime. The CURV discriminator tends to break up large features
into multiple small features, and large features tend to have wider wings than the Gaussian profile that is
assumed by CURV, lowering the number of detections above \(2.5 \times 10^{18}\) Mx. YAFTA and SWAMIS/down
also tend to break up very large concentrations of flux into multiple features, but that effect is not as strongly
apparent in the derived size distribution below fluxes of \(10^{19}\)Mx. YAFTA and SWAMIS agree quite well
on the size distribution from the YAFTA threshold to several \(\times 10^{19}\)Mx.

In the moderate-strength feature range of \(2 - 5 \times 10^{18}\) Mx, all four of YAFTA, MCAT, SWAMIS/down,
and SWAMIS/clump are in reasonably good agreement, with the main difference being between the downhill-
like codes (CURV, YAFTA, and SWAMIS/down) and the clumping codes (MCAT, SWAMIS/clump). The
difference is due to the segmentation of large features into several smaller ones, giving the downhill-like
codes slightly more small features and slightly fewer large ones.

The different codes disagree substantially on slope of the size distribution curve in two different re-
gions. Because of its more sensitive discriminator, CURV finds many more small features than YAFTA or
SWAMIS, yielding a sharper slope at the small end of the distribution; but in the window of \(1.5 - 5 \times
10^{18}\) Mx, all three codes agree on the slope of \(-0.28 \pm 0.3\) decade \((10^{18}\text{Mx})^{-1}\), or an e-folding width of
\(1.55 \pm 0.15 \times 10^{18}\) Mx. The downhill-like methods find the steeper limit and the clumping methods find
the shallower limit. Features in this size range are strong enough to be detected by all three discriminators
but not so large that the differences between the large-scale behavior of the three codes is important.

5. Discussion

Each of the techniques we considered has advantages for a particular regime of fragment size and
strength relative to the noise floor of the instrument. Here, we discuss the tradeoffs of the different detection
schemes. The main differences between our codes lay in the discrimination and feature-identification steps,
which are discussed separately.

5.1. Discrimination

The main problem faced by tracking discriminators is the huge number of statistically independent
samples across an image sequence dataset. The simplest discriminator is a threshold trigger; while thresh-
old triggers are inadequate when used alone, they form the basis of every discrimination algorithm. The
three main ways we improved upon simple threshold triggering were curvature sensing (CURV), hysteresis
(MCAT, SWAMIS), and post-discrimination filtering for feature size and longevity (all codes). One code
(YAFTA) was optimized for strong field detections and used simple trigger discrimination.

Curvature sensing as implemented in CURV has the advantage that, when combined with a threshold
trigger, it applies five statistically independent threshold tests to each pixel, significantly reducing the false-
positive rate. CURV rejects features whose convex cores are smaller than 9 pixels. Including the effects of
smoothing in the preprocessing steps, which leave granulation as the dominant source of noise, there are
about 12 statistically independent tests (of 45 total conditions) are required to detect a particular feature. By contrast, a direct trigger yields only about three statistically independent tests with the same size threshold. This permits a detection threshold much closer to the noise floor than would otherwise be possible, which in turn makes curvature discrimination rather sensitive to small scale, weak concentrations of magnetic flux.

The disadvantage of curvature discrimination is that it only finds the convex core of a magnetic feature. This is addressed by Hagenaar et al. (1999) via a simple scaling: they find that for a large variety of near-Gaussian distributions the convex core is about 1/3 of the total flux in the feature, and scale accordingly. This works well for small features near the resolution limit of the observations, but not as well for larger features, which are observed to have flatter profiles than a Gaussian. Large concentrations of flux typically have several local maxima, and the total flux may be over- or underestimated by the assumption of a simple Gaussian shape, depending on the actual morphology of the feature.

Hysteresis is a simple way of reducing the false positive rate of threshold-trigger discrimination. Pixels are compared against different trigger thresholds depending on whether they are isolated or adjacent to other detected pixels. Both MCAT and SWAMIS use a recursive-hysteresis scheme that starts with a simple threshold scheme, and then dilates detected pixels using a lower threshold. Such schemes eliminate many false positives due to the background noise floor, and detect the full extent and shape of large features. The drawback is that weak features are only detected if they have at least one 'seed' pixel that is stronger than the high threshold. SWAMIS further allows dilation along the time axis, so that weak features are detected if at some point in their lifetime they have a single strong pixel; but even so many transient weak features go undetected for lack of a single strong pixel.

### 5.2. Feature Identification

Here, we contrast the two principal dilation strategies of the codes: downhill and clumping dilation from local maxima. The distinction between these strategies is academic within CURV, as the curvature-based discriminator provides well-separated loci around each local maximum: the final detected loci are the same regardless of dilation method. MCAT, SWAMIS, and YAFTA can dilate using clumping, and SWAMIS and YAFTA can dilate using the downhill technique.

Both the downhill and clumping techniques, together with hysteretic thresholding, do better than curvature at identifying the size and shape of mid-sized magnetic features in the several arc second size range. In this size range the shape of individual features varies considerably, though most features still have but one local maximum in the MDI data that we tracked; under these conditions, both dilation techniques do about as well as one another and both measure the flux of individual features with more precision than CURV (which uses a simple geometric factor to estimate the flux in the wings of the structure).

Large scale structures that are more than about 15 arc seconds across yield stronger differences between the downhill and clumping techniques, as illustrated in Figure 6. The downhill technique does a better job at tracking substructure of large, extended objects such as plage and active region fields, but at the expense of more noise susceptibility. Because small amounts of noise can produce transient local maxima in a large
Fig. 6.— Effect of feature identification technique: a pathological case. Clumping (right) is less sensitive to noise in weak features than is downhill dilation (left), but can lead to counterintuitive results in active regions as large, irregular patches of flux are identified as a single object.

extended feature, the downhill technique is susceptible to the *swiss cheese problem* in which a single large clump of flux with no strong local maximum can be oscillate between being detected as one or several separate features. If lifetime filtering is being applied to the detected-feature list, then large holes may appear in the detected feature, giving it the appearance of an irregular block of Emthaler. Furthermore, downhill detection alone tends to miss very large concentrations of flux, treating them as a collection of smaller features: while this is desirable for tracking the motion of the solar surface, it is not desirable when measuring the statistics of sizes of magnetic features.

We discuss the tradeoff between these detection techniques, neither of which is perfect, in §6, below.

### 5.3. Cross-frame feature association

All of the codes we compared use essentially the same cross-frame association strategy of finding the association map that maximizes overlap between features in adjacent frames, as described in §2.4. In practice, most features in most frames overlap with exactly one feature in the following and adjacent frames, so variations in the type of overlap (e.g. number of pixels vs. amount of flux) or of permissiveness of overlap (e.g. including pixels nearby each feature as part of the feature itself, for purposes of finding overlap) only affect the “edge cases” in which multiple magnetic features are interacting, or in which a single feature is moving rapidly.

Overlap-style algorithms such as described in 2.4§ are quite robust for associating features with the properties \( \forall i, j : |\vec{x}_i - \vec{x}_j| \gtrsim r_i \) and \( \forall i : |\Delta \vec{x}_i| \lesssim r_i/2 \), where \( i \) and \( j \) are indices across features, \( \vec{x}_i \) is the centroid location of feature \( i \), \( r_i \) is the typical radius of feature \( i \), and \( \Delta \vec{x}_i \) is the displacement vector of feature \( i \) across frames. Fast moving features with \( |\Delta \vec{x}_i| \gtrsim r_i/2 \) become subject to the mistaken identity problem, where they are identified as a different feature in different frames. The mistaken identity problem affects statistical feature-lifetime and feature history results even if only a very few features are subject to it, because a single fast-moving feature may register as a very large number of separate magnetic features. The only reliable way to beat the mistaken identity problem is to use high enough time resolution in the
data. Marginal data in which the fastest moving features have $\Delta x_i \approx r_i$ may be improved by interpolating interstitial frames, but wider separations cannot be helped by that method. In practice, we found by visual inspection that 12-minute final effective cadence with direct boxcar averaging was not sufficient to avoid the mistaken identity problem for the fastest moving features at the MDI “full disk” resolution (1.4 Mm pixels at Sun center), accounting for ~0.5% of features in a given frame and perhaps 5% of total feature IDs assigned by SWAMIS, but that 12 minute final effective cadence with anti-aliasing in the time direction (time-weighted averaging of 1 minute cadence magnetograms, with a 12 minute FWHM Gaussian weighting profile) eliminates virtually all cases of mistaken identity.

We considered, but did not implement, various methods to reduce the mistaken identity problem in cases where high enough cadence data are not available. Promising directions to try include linear extrapolation of feature location from last associated location before the overlap calculation, with or without backtracking on the time axis; dilation with size checking; and simulated annealing of feature association. We considered that all such algorithms are likely to include more faulty associations than does direct overlap.

### 6. Recommendations on Accepted Practice

To enable meaningful comparison and reproducibility of tracking results across research groups, we recommend the following techniques as appropriate for most applications of magnetic feature tracking:

#### 6.1. Data preprocessing

While preprocessing of data is not technically a part of feature tracking, preprocessing can affect the statistics of image tracking and therefore warrants mention here. We discuss despiking, time averaging, and resampling into a desired coordinate system.

**Despiking** A brief note on space-based magnetograms is in order: *SOHO*/MDI is, and presumably *SDO*/HMI will be, susceptible to cosmic ray impacts. A typical MDI “full-disk” magnetogram has evidence of cosmic ray impacts in $10^2$ pixels, so 5-10 minute averages may have as many as $10^3$ bad pixels caused by cosmic rays. The cosmic rays are not saturated in the images, and may have either negative-going or positive-going direction. These cosmic rays can skew the size and lifetime statistics of small, short-lived features if not considered. We recommend either despiking sequences of space-based magnetograms with a second time derivative technique such as ZSPIKE (DeForest et al. 2004), or imposing a lifetime threshold on detected features to limit the effects of cosmic rays.

**Time and spatial averaging** Time averaging of images is useful as a preparatory step to reduce noise in the magnetograms and to smooth features for better association across frames. There are several sources of noise in currently available magnetograms, with different statistics for each source; we discuss them briefly
here, as the noise characteristics of averaged data sets hold a complex relationship to the noise characteristics of individual frames.

Most magnetographs are photon-limited, so that there is an approximately Gaussian distribution noise source of photon noise associated with photon counting statistics in each pixel of each magnetogram. Each pixel contains an independent sample of this noise source. Magnetographs such as MDI that assemble multiple exposures are subject to shutter noise, which results from very slightly different exposure times across each independent exposure used to produce the magnetogram: shutter noise is an approximately Gaussian distribution noise source that is added to the common mode of all pixels across each image. Finally, solar evolution (and, for ground-based telescopes, seeing effects) across the time of assembly of the magnetogram induces an additional noise source, evolution noise that is dominated by the evolution of granules. Granulation, and the associated evolution noise, has about one independent sample every five minutes, per square megameter of solar surface area.

Individual MDI magnetograms have about equal amounts of photon and evolution noise. Because the photon noise is independently sampled in each image, averages of more than about five minutes of magnetic data tend to be dominated by evolution noise, which is attenuated much more slowly by further averaging.

Anti-aliased time averaging, using overlapping Gaussian or Hanning windows in the time domain, is preferable to simple boxcar averaging, which has frequency sidelobes that allow more noise to enter the data.

**Image resampling** Image sequences are typically resampled to remove the solar rotation and perspective *a priori*. For the present work, we derotated and prepared time averages of MDI magnetograms using a simple interpolation scheme into plate caree coordinates. This scheme follows current common practice but is not recommended: it fails to preserve small-scale feature statistics in two important ways.

First, the plate caree (“lon/lat”) map projection is non-authalic: a feature of unit area on the surface of the Sun may have different areas in plate caree coordinates, depending on its latitude. To avoid skewing the statistics of flux content, authors should use an authalic (equal-area) projection to prepare the data before tracking. The area of a feature in the plate caree projection is scaled by a factor of secant(longitude). A simple way to compensate is to scale the vertical or horizontal scale by cos(latitude) at each point. Scaling the vertical axis by cos(latitude) yields the common *sin-lat cylindrical projection*, so named because integrating the scale factor $y_{\text{map}} \cos(\text{lat})$ yields $y_{\text{map}} \sin(\text{lat})$. Scaling the horizontal axis yields the *sinusoidal projection*. Other useful authalic choices include the Hammer/Aitoff elliptical projection used by the cosmology community and Lambert’s azimuthal equal-area projection, which minimizes linear distortion near the origin. Many useful projections have been cataloged by Snyder (1987).

Secondly, linear interpolation leaves much to be desired as a resampling method, skewing (among other things) the noise profile of individual pixels and potentially introducing large amounts of distortion into the statistics of small features. A statistically sound, photometrically accurate resampling method, relying on spatially variable sampling filters, has been described by DeForest (2004); that or similar techniques are recommended for preparing data for survey applications.
For virtually every application of tracking, it is important to compensate by rigid rotation based on the differential rotation speed at a particular point in the field of view, and not by differentially rotating every pixel in the image independently. The former preserves the actual evolving spatial structures in question; the latter only preserves the plasma reference frame at the start of the observing run.

6.2. Feature discrimination & identification

We recommend combining the three methods of feature detection. Standard codes should use a dual-discriminator scheme for detection: an initial convex-core discrimination as in the Hagenaar code, followed by dilation to a low noise threshold. This combination takes best advantage of the extra discrimination afforded by the convex core technique, while eliminating some of the difficulties of identifying oddly-shaped and large features.

Feature identification should use the downhill method to avoid pathologies of the clumping technique, particularly when used for motion tracking and to identify interacting magnetic features; but for applications where larger clusters of flux are important, we recommend keeping track of groups of touching or nearby features according to a clumping algorithm. Groups of mutually touching features are the same loci as would be identified by a direct clumping scheme, but tracking individual peaks within the group affords better localization of the magnetic flux that makes up the feature(s).

6.3. Feature association

For best general purpose utility, we recommend a flux-weighted maximum overlap method of association between frames, as is currently used by SWAMIS; for example, in cases of associative conflict such as Figure 2, regions B and C would be associated as identical, region A would be classified as dying by merger into B/C, and region D would be classified as originating by fragmentation from B/C. For analyses that require feature identification, it is important to ensure that the cadence is sufficient to allow associated features in adjacent frames to overlap.

6.4. Feature tabulation

When tabulating feature histories, we recommend that the following minimum information be kept for each feature, and for each frame for which a particular feature exists: area (A), total flux (Φ), flux-weighted average location (x,y), and flux-weighted quadrupole moments (< Φ^2dx^2 >, < Φ^2dy^2 >, < Φ^2dxdy >), for a total of 7 numerical quantities per feature per frame. The quadrupole moments, in particular, summarize tersely and simply the shape of the feature, and the features detected by the downhill dilation method tend to be simple shapes that are readily described with the quadrupole moment set. Quantities may be kept in physical or image units (e.g. km or pixels). Quantities which we recommend avoiding are: pixel value
maximum and variance, which depend on resolution and phase of the underlying feature relative to the pixel grid; and non-weighted average location, because it is more dependent on noise-dominated pixels at the feature’s edge than is the flux-weighted average location.

6.5. Event identification

Several of the scientific applications of tracking require classifying the origin and demise of each feature based on visual heuristics for the underlying physics. Useful event classification requires characterizing the geometry and manner of change of nearby features. Event classification is a rich topic that is not fully discussed in this paper; however, we make some brief recommendations.

We recommend classifying origin events into four categories: (i) isolated appearance, in which a particular feature appears in the absence of interaction with surrounding detected features; (ii) balanced emergence, in which a bipolar, approximately balanced pair of features appear together in nearly the same location at nearly the same time; (iii) unbalanced emergence, in which a new feature appears next to a pre-existing, opposite sign feature in a nearly flux-conserving manner; and (iv) fragmentation (or splitting), in which a single pre-existing feature breaks up into multiple smaller features in a nearly flux-conserving manner. Demise events should be classified in the exact same way as origin events, in a time reversed sense: (i) isolated disappearance; (ii) balanced cancellation; (iii) unbalanced cancellation; and (iv) merging. For completeness, event identification software should also maintain a complex class for events which cannot be classified easily into the above four groups.

It is important to understand that this is a visual classification scheme, to be more fully developed in future work. Interpretation of these visual events in terms of physical mechanisms is neither straightforward nor obvious. For example, appearance events may or may not correspond to new flux on the solar surface; this topic is explored in more detail in Hагенаар et al. (Paper II).

Results using our recommended classification scheme should be presented with a notation describing what criteria are used to detect balanced changes in the flux of interacting features. Event classification results can be quite different, for example, if the $\Delta \phi_i$ of two interacting features is considered “approximately balanced” if they merely share the same sign, or if they must agree within, say, 10%.

7. Conclusions

We have compared four magnetic feature tracking codes by applying them to the same preprocessed set of magnetic data. Feature tracking code output is sensitive to a variety of decisions that are made during development, and this sensitivity is a reason why it has historically been difficult to reproduce results obtained by feature tracking: it is crucial to explain exactly what algorithm is being used. In particular, codes that were designed for one regime of study (e.g. very small intranetwork flux concentrations or very large, strong features) should not be applied to different regimes of detection without careful study, and all
discrimination and association techniques need to be laid out exactly as performed.

To aid future scientific work, all four of the codes are being made available to the scientific community in source-code form via solarsoft.

By comparing the algorithms, we have determined why the codes produce different results for the flux distribution in quiet sun, and evaluated under what circumstances each technique performs best. Further, we have made recommendations about how to improve feature detection and reproducibility in feature tracking for future work.
REFERENCES


**Glossary**

Feature tracking and magnetic observations are mature enough to have developed a collection of commonly used terms, which unfortunately have drifted into slightly different usage in different locations. In an attempt to regularize terminology, we present a glossary of commonly used terms, with their recommended definitions. Also, because some terms are strictly observational and others imply a physical model, we have noted which are which.
Object descriptions

bipole - a pair of magnetic features of opposite sign and approximately equal flux content, that appear to be associated (as in bipolar emergence). When seen to emerge together, the poles of a bipole may be associated observationally.

ephemeral region - a resolved small bipole with particular properties (which are...?).

feature - a visually identifiable part of an image, such as a clump of magnetic flux or a blob in a magnetogram. The term “feature” is purely observational and is preferable to “flux concentration” when describing individual visual objects in an image. The specific definition of a feature is dependent on both the Sun itself and the characteristics of the observing telescope.

flux concentration - a localized cluster of magnetic flux, with or without resolved substructure. A flux concentration may consist of one or more magnetic features. While somewhat vague, the definition of a flux concentration is approximately independent of observing telescope: a flux concentration may appear as a single feature when seen with one instrument but as several features with another.

fragment - a small piece of a larger magnetic structure, not a generic small bit of magnetic flux. Usage: “this magnetic flux concentration is composed of many fragments”, or “unresolved fragments make up this magnetic feature”. “Fragment” should not be used interchangeably with “feature”, as it implies that the subject is part of a larger whole, while “feature” does not.

monopole - a lone magnetic pole (thought to be physically impossible).

stenflo - (after J. Stenflo) a tiny, strong concentration of order $10^{17} \text{ Mx}$ of flux. Usage: “The asymmetric formation of flux concentrations in the network may be due to convergence of stenflos, though Stenflo may object to this terminology.”

unipole - a single magnetic feature with no obvious associated feature of the opposite sign (oppose bipole, monopole). The photospheric boundary provides a “hiding place” for the opposing pole, so that unipoles are thought not to be monopoles.

Event descriptions

appearance - used specifically to describe the origin of a single unipolar feature where there were none before. Appearances appear to violate conservation of magnetic flux, but probably result from flux hiding under the noise floor of an instrument – so the definition of “appearance” depends on the instrument being used.

calving - a form of fragmentation in which one of the daughter features contains much more flux than the other, by analogy to the behavior of icebergs. Usage: “This movie shows small features calving off of the main flux concentration”. (oppose “splitting”; compare “fragmentation”)
cancellation - the demise of a magnetic feature that collides (and cancels) with an opposing sign feature, in such a way that flux is approximately conserved.

coalessence - the collection of diffuse flux from below detection threshold to a small, denser feature that can be detected. This may be an example of unresolved merging. This is a physical description of one type of feature appearance; asymmetric bipolar emergence is another type. To avoid confusion, eschew “coalescence” when describing observational results; use “merging” or “appearance” instead.

demise - the end of a magnetic feature’s existence.

disappearance - the end of a single, unipolar magnetic feature that “fades away” to nothing in the absence of nearby features (the time reversal of an “appearance”).

dispersal - deprecated. This has been used to describe the opposite of coalescence, the breakup of strong flux concentrations into many fragments, and the diffusion of flux across the surface of the Sun. It is now too ambiguous to be used clearly in most cases.

emergence - the origination of two balanced, opposing magnetic features nearby one another in such a way that flux is approximately conserved. This observational definition follows the common physical definition of a loop of flux emerging from below the surface. Isolated unipolar structures originate in “appearances”.

fragmentation - the breakup of a single magnetic feature into at least two like-sign features. (compare “splitting”, “calving”)

merging - the joining of two magnetic features of similar sign into a single larger feature.

splitting - the breakup of a single magnetic feature into at least two like-sign features, with the implication of rough flux balance between the two daughter features. (oppose “calving”; compare “fragmentation”).

Thanks to the SOHO/MDI team for kind use of their data, and to the University of St. Andrews for hosting the workshop which made this comparison possible. This work was funded by NASA’s SOHO project, the SOHO/MDI effort, NASA’s SEC-GI program, the Air Force Office of Scientific Research MURI program, and the PPARC Advanced Fellowship program. SOHO is a project of international collaboration between NASA and ESA.