

An evolutionary computation based algorithm for calculating solar differential rotation by automatic tracking of coronal bright points

Ehsan Shahamatnia^{1,2,*}, Ivan Dorotovič^{1,3}, Jose M. Fonseca^{1,2}, and Rita A. Ribeiro¹

¹ Computational Intelligence Group of CTS/UNINOVA, 2829-516 Monte de Caparica, Portugal

*Corresponding author: ehsan.shahamat@gmail.com

² Department of Computer and Electrical Engineering, FCT/NOVA University of Lisbon, 2829-516 Monte de Caparica, Portugal

³ Slovak Central Observatory, Hurbanovo, Slovak Republic

Received 31 January 2015 / Accepted 2 February 2016

ABSTRACT

Developing specialized software tools is essential to support studies of solar activity evolution. With new space missions such as Solar Dynamics Observatory (SDO), solar images are being produced in unprecedented volumes. To capitalize on that huge data availability, the scientific community needs a new generation of software tools for automatic and efficient data processing. In this paper a prototype of a modular framework for solar feature detection, characterization, and tracking is presented. To develop an efficient system capable of automatic solar feature tracking and measuring, a hybrid approach combining specialized image processing, evolutionary optimization, and soft computing algorithms is being followed. The specialized hybrid algorithm for tracking solar features allows automatic feature tracking while gathering characterization details about the tracked features. The hybrid algorithm takes advantages of the snake model, a specialized image processing algorithm widely used in applications such as boundary delineation, image segmentation, and object tracking. Further, it exploits the flexibility and efficiency of Particle Swarm Optimization (PSO), a stochastic population based optimization algorithm. PSO has been used successfully in a wide range of applications including combinatorial optimization, control, clustering, robotics, scheduling, and image processing and video analysis applications. The proposed tool, denoted PSO-Snake model, was already successfully tested in other works for tracking sunspots and coronal bright points. In this work, we discuss the application of the PSO-Snake algorithm for calculating the sidereal rotational angular velocity of the solar corona. To validate the results we compare them with published manual results performed by an expert.

Key words. Sun – Helioinformatics – Corona – Solar image processing – Machine learning

1. Introduction

To develop an automatic solar feature tracking system, a hybrid approach combining an image processing technique and an evolutionary computation algorithm is used. The evolutionary computation algorithm, Particle Swarm Optimization (PSO), is a search algorithm for finding the global optimum, inspired by the social behavior of bird flocking (Kennedy & Eberhart 1995). The original idea of PSO is to simulate how birds interact with each other and with their environment in search for food. That makes PSO a population-based evolutionary computation algorithm consisting of relatively simple individuals, called particles, where each particle represents a potential solution to the optimization problem at hand. PSO tries to find the optimum by iteratively improving the candidate solutions in regard to a given measure of quality. In the search for optimal solutions, particles keep track of their best values and communicate the best solutions, per iteration, back to the population. Instead of exhaustively searching the whole search space, PSO evaluates only a small portion of the search space. The particle swarm searches more efficiently by paying attention to the more promising areas of the search space. The search is conducted in a combination of stochastic manner with some guiding mechanisms. Since the algorithm uses primitive mathematical operations it is computationally inexpensive to implement. The simplicity, flexibility, and good performance of PSO have made it a popular choice as a global problem

solver in a wide range of real-world applications such as human tremor analysis (Eberhart & Hu 1999), tracking dynamic systems (Eberhart & Shi 2001), RNA molecule structure prediction (Agrawal & Agrawal 2015), and synthesis of antenna arrays (Ram et al. 2014). In many applications where PSO has been used, it has shown consistently good performance (Hu et al. 2004; Poli 2008; Yang 2015). Moreover, thanks to its speed, simplicity, and flexibility in formulating problems, PSO has been successfully used in many hybrid algorithms to solve specific problems such as antenna optimization, classification of biological data, and vehicle routing (Robinson et al. 2002; Holden & Freitas 2005; Marinakis et al. 2010).

PSO is a member of the family of stochastic optimization methods, along with genetic algorithms and simulated annealing, two other popular algorithms in this field. That type of optimization algorithm does not guarantee finding an optimal solution, but these algorithms are good at finding near optimum solutions, by searching very large spaces and making few or no assumptions about the problem. Although the choice of algorithm is usually affected by the problem specifications, several studies in the literature show that PSO outperforms genetic algorithms and simulated annealing in many test problems investigated (Habib & Al-kazemi 2005; Hassan et al. 2005; Panda & Padhy 2008; Yang et al. 2008; Ethni et al. 2009). Further, it is reported (Hassan et al. 2005) that the convergence rate of PSO is steadier than genetic algorithms and

that its computational efficiency and execution time are better because of the smaller number of function evaluations needed for convergence. Unlike classic optimization methods, such as gradient descent and quasi-Newton methods, PSO does not use the gradient, therefore it can also be used for optimization problems that are not differentiable, are partially irregular, noisy, or change over time.

A specialized image processing technique is included in the hybrid PSO-Snake algorithm. Image segmentation is a frequently addressed problem in digital image processing and deformable contours are well-known algorithms for object boundary delineation. Introduced by [Kass et al. \(1988\)](#), the active contour model is a class of deformable contours for finding features of interest in an image, by formalizing it as an optimization problem. Starting by a rough approximation of the object boundary, contours evolve to find the precise boundary of that object ([Ballerini 1999](#); [Ballerini & Bocchi 2003](#)). Evolving the contour takes into account the low-level image data such as image gradient as well as high-level image information such as contour continuity, shape, texture, color, etc. The active contour model is also known as the “snake model” and the contours in this model are called snakes. Due to their flexibility, snakes are widely used in several applications such as image segmentation, shape modeling, stereo matching, and object tracking ([Ballerini & Bocchi 2003](#); [Karlsson et al. 2003](#); [Niu 2006](#); [Wildenauer et al. 2006](#)). For every snake there is an energy function associated with it. Specifically, the snake model is an optimization algorithm that works as an energy minimization procedure where the snake with the lowest energy is considered the best match for representing the object. The snake’s evolution process starts by setting up an initial curve (snake) and moving it toward the target object. The snake movement is governed both by internal forces, within the curve, and external forces from the image. The former maintains the snake shape while the latter steers the snake toward the image feature, defined by the external forces. Snake energy is calculated as a combination of those two forces. In the standard snake model, this energy is minimized in a numerical way by iteratively solving a pair of Euler equations. This is usually a computationally expensive operation.

Other problems associated with the snake model are sensitivity in placing the initial snake and the poor convergence at concave object boundaries ([Davatzikos & Prince 1994](#); [Bresson et al. 2007](#)). Many researchers have tried to ameliorate this problem by improving the capture range of image forces ([Cohen & Cohen 1993](#); [Leroy et al. 1996](#); [Prince 1997](#); [Park et al. 2001](#); [B. Li & Acton 2007](#)). Several works ([Amini et al. 1988](#); [Mun et al. 2004](#); [Bresson et al. 2007](#)) have tried to address the snake model limitations in regard to noise tolerance, local minima sensitivity, and stability. Very few have succeeded in proposing an alternative that would solve the snake model’s limitations without compromising its performance, flexibility, and simplicity.

One of the successful variations of the snake model is the geometric active contour model, particularly popular for medical image analysis ([McInerney & Terzopoulos 1996](#)). Geometric snakes, sometimes called geodesic snakes ([Caselles et al. 1997](#); [Paragios & Deriche 2000](#); [Xu et al. 2000](#); [He et al. 2008](#)), are mainly implemented based on the level-set theory to capture boundaries through a continuous curve, and thus are able to adapt to topology changes while evolving contours. With their intrinsic capability to manage splitting and merging contours, they can be used to overcome the sensitivity to the initial snake configuration. In its standard implementation,

the level-set function of all pixels in the image is updated at each iteration, which makes the geometric active contour model computationally intensive. There have been suggestions for lowering the computational complexity of this method, but issues related to noise and ill-defined boundaries (both present in the solar images) remain problematic. Thus, geometric snakes are not further discussed in the scope of the current work.

In the standard snake model, the contour is defined by a set of finite points called control points. For evolving the initial contour toward the final object boundaries, snake energy is minimized, i.e. snake control points are iteratively updated by solving a pair of Euler equations. Other methods to minimize the snake energy have been suggested such as: dynamic programming ([Amini et al. 1988](#)), greedy algorithms ([Lam & Yan 1994](#)), genetic algorithms ([Ballerini 1999](#); [Ballerini & Bocchi 2003](#); [Mun et al. 2004](#)), and swarm-based optimization algorithms ([Asl & Seyedin 2006](#); [Zeng & Zhou 2008](#); [Nebti & Meshoul 2009](#); [R. Li et al. 2009](#); [Tseng et al. 2009](#); [Shahamatnia & Ebadzadeh 2011](#)). In recent years, several researchers have used PSO to optimize the snake energy, mostly by constraining the search space. Some works have used multi-population PSO ([R. Li et al. 2009](#); [Tseng et al. 2009](#)) where every control point is confined to a sub-swarm spatially distinct from other sub-swarms. [Nebti & Meshoul \(2009\)](#) restrict the search space of each contour control point using polar coordinates. [Zeng & Zhou \(2008\)](#) use an iterative method to rank the best set of particles’ position at each epoch, preventing particles from intersecting.

Most of those methods are used as a general optimization technique in solving the snake model equations without modifying the snake structure. They formulate the snake energy calculations as a minimization problem according to each method’s specifications. In this paper, we take the hybrid PSO-Snake approach introduced in [Shahamatnia & Ebadzadeh \(2011\)](#) and explore its versatility by further extending it to solve a real-world problem from the astrophysics domain. The hybrid PSO-Snake model is no longer a general problem solver, but it is a specialized image processing technique for searching the image space. The method presented here customizes the previous PSO algorithm ([Shahamatnia & Ebadzadeh 2011](#)) to overcome some snake model drawbacks including snake initialization, concave boundaries, sensitivity to noise, and local minima. The performance of the PSO-Snake algorithm regarding those issues was measured using synthetic images, specifically generated to isolate each issue, and the results are discussed in detail by [Shahamatnia & Ebadzadeh \(2011\)](#).

In the PSO-Snake model the simple structure of the PSO is preserved yielding an algorithm with a low order of complexity and hence good processing time. These factors are of utmost importance for precisely calculating the differential rotation of solar features. Determining the exact nature of the differential rotation of both the solar surface and the solar interior is still one of the most serious open issues of solar physics ([Thompson et al. 1996](#); [Scherrer et al. 2011](#); [Hanasoge et al. 2015](#)). The solar surface rotates differentially, lower latitudes rotate faster than higher latitudes ([Howard 1984](#)). However, the differential rotation mechanism, most likely caused by the interactions between the convection and the overall rotation, is not exactly known. Differential rotation plays an important role in solar activity generation – at least the large-scale manifestations of solar activity are related to changes in the local magnetic field which may have their roots in variations

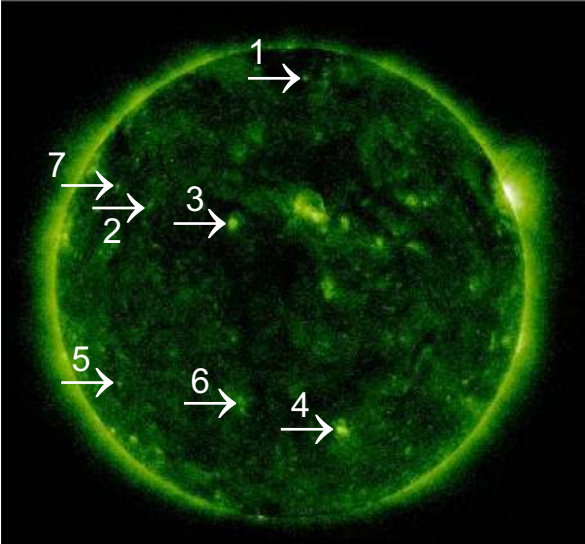


Fig. 1. A sample solar image from the 5th October 2010 with several CBPs marked. (Image courtesy of NASA/SDO).

in the differential rotation. Rotational irregularities may also serve as indicators of hypothetical processes beneath the solar surface. One example could be the location of a layer where rotational speed changes abruptly (a jet stream). Sometimes it is called a layer of torsional oscillation (Howard & LaBonte 1980). The current paper focuses on reporting the results of applying the proposed tool on coronal bright points (CBPs). In this work we trace the location of bright points in a series of selected images obtained by the Atmospheric Imaging Assembly (AIA; Lemen et al. 2012), an instrument on board the Solar Dynamics Observatory (SDO; Pesnell et al. 2012).

Coronal bright points (CBPs) or simply bright points are small and bright structures observed in the extreme ultraviolet (EUV) and in the X-ray frequencies of the solar spectrum (Habbal & Withbroe 1981; Brajša et al. 2001). They are known to have a mean lifetime of about 8 h, a typical maximum area of $2 \times 10^8 \text{ km}^2$, but still they look like a small dynamic loop structure on the solar images (Brown et al. 2001) as shown in Figure 1. Bright points are associated with small, bipolar magnetic features in the photosphere. A large quantity of bright points (several thousand) emerges over the surface of the Sun each day.

Tracking the coronal bright points with high precision, over extended periods of time, will help solar physicists and space weather scientists to better understand this important solar feature. Such automatic tools will allow solar researchers to precisely process large amounts of solar data and hence improve solar physics models. Hence, the main aim of this paper is to assess the results of applying a hybrid PSO-Snake algorithm to tracking of coronal bright points. Due to the dynamic nature of PSO-Snake hybrid algorithm, detected contours are flexible and can conform to changes in shape and size of the deformable objects like CBPs. The tracking result is then used for calculating the differential rotation of coronal bright points. The result of PSO-Snake hybrid algorithm is cross-referenced and compared with a state-of-the-art study, which entails a manual procedure done by an expert (Lorenz et al. 2012).

Small CBPs were chosen for tracking because they were well detectable and suitable for the PSO-Snake (or any other) tracking algorithm. Compared to other solar features such as

sunspots, CBPs have simpler shapes and smaller structures. CBPs are particularly suitable for the PSO-Snake algorithm, because of their small size and their shape that make them visually distinguishable. This facilitates the convergence of the initial snake to the boundaries of the CBP in the initial phase of the PSO-Snake algorithm. In this study, for calculating the velocity, the CBP is represented by a single reference point. In the case of CBPs this point can be considered as the center of mass of the structure, while for complex shapes such as some sunspots such assumption is not intuitive. Another characteristic of the CBPs that makes them suitable for the PSO-Snake algorithm is that they are usually spatially confined, contrary to some other solar features such as filaments. The PSO-Snake algorithm is a contour-based tracking scheme and while it is well capable of adapting to deformities, changing shapes, and recovering from partial edge information, it cannot overcome too many faint borders, completely disconnected object segments, and fine details of the complex shapes.

CBPs are also very good tracers since they extend to much higher latitudes than sunspots (Sudar et al. 2015). They are also one of the solar corona's most ubiquitous features (McIntosh & Gurman 2005), quite numerous in all phases of the solar cycle while, for example, sunspots are often absent in the minimum of the cycle (Sudar et al. 2015). McIntosh & Gurman (2005) developed a method of automatically detecting EUV bright points and applied it to the archive of EIT data from the launch of SOHO in 2005. As a result, they produced a database of all detected bright points that can be used to extract numerous diagnostics of the solar corona over the 23rd solar cycle. CBPs are also suitable tracers for the determination of the solar differential rotation, because they are localized objects which are very well distributed over the solar disk (Brajša et al. 2014). The rest of this paper is organized as follows: Snake model, PSO, and PSO-Snake algorithms are reviewed in Section 2. Section 3 provides the experimental results and discussions. Finally, conclusions are provided in Section 4.

2. PSO-Snake hybrid algorithms

The first version of the hybrid algorithm is a merger of the snake model and PSO (Shahamatnia & Ebadzadeh 2011). It integrates the snake model's contour evolving paradigms with particle dynamics from PSO. After that the region of interest is approximated and the snake model will be able to find the precise boundary of that object.

As mentioned before, in the snake model, the contour of snake has an energy associated with it that is related to the location of the snake on the image (external force) and with the geometrical characteristics of the snake (internal force). The idea is to minimize the integral measure that represents the snake's total energy, by evolving the snake over time. The original snake model achieves this minimization by iteratively solving a pair of Euler equations on the discrete grid, resulting in a computationally expensive algorithm (Karlsson et al. 2003). Two main approaches for snake representation are geometric and parametric representations. Geometric models use an implicit representation based on the curve evolution theory and are usually implemented with level-set techniques. Effectively handling multiple objects and topology alteration is the advantage of this approach, with the cost of being computationally more complex. On the other hand, the parametric approach is computationally efficient and easy to interact with

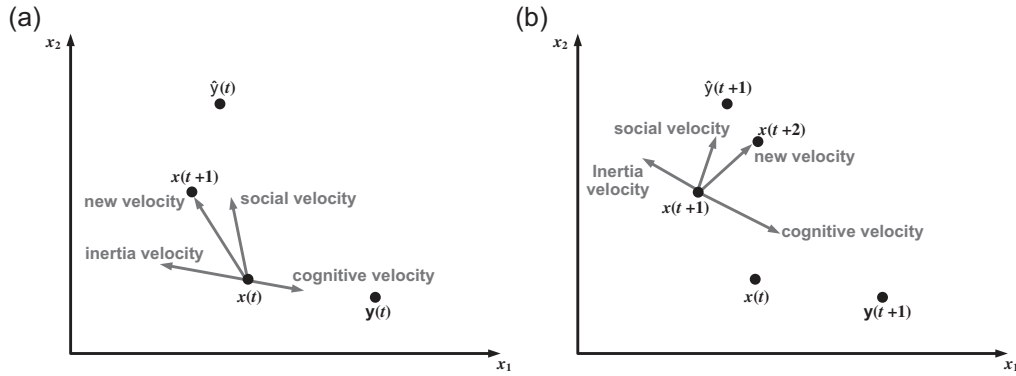


Fig. 2. Search mechanism of PSO particle in a two-dimensional space. (a) Shows particle x at time t under influence of its inertia, cognitive and social velocity components, (b) new position of the particle at time $t + 1$ is the algebraic sum of the velocities acting upon it.

users (Hornig et al. 2010). In the parametric implementations, the snake is defined as a curve $p(s) = (x(s), y(s))$, having arc length s . As it is shown by Eq. (1), a number of discrete points, called control points or snaxels, characterize the snake at each given time, t (Kass et al. 1988). The PSO-Snake hybrid uses this representation since it matches the snaxels to the PSO particles well. The total snake energy is the sum of its internal (spatial) and external (geometrical) integrals as shown in Eq. (2).

$$p(s, t) = (x(s, t), y(s, t)), \quad s \in [0, 1], \quad (1)$$

$$E_{\text{snake}} = \int_0^1 E_{\text{int}}(p(s)) ds + \int_0^1 E_{\text{ext}}(p(s)) ds. \quad (2)$$

The snake model is considered to be a controlled continuity spline under the influence of internal and external forces that induce the snake energy. Internal energy, shown in Eq. (3), consists of two terms that are the first and the second derivatives of the snake with respect to s . The first term coerces the spline to act like a membrane while the second term makes the snake to act like a thin plate (Kass et al. 1988). The external energy determines the snake's relationship to the image. It is formulated in a way that its local minima correspond to the image features of interest. Various external energies can be employed such as image intensity, image gradient, object size or shape. One common definition used for gray-level images, $I(x, y)$, is the gradient of Gaussian, as shown in Eq. (4), where γ is the tuning coefficient and ∇ is the operator to calculate the gradient of the resulting image. $G_\sigma(x, y)$ is a two-dimensional Gaussian function with the standard deviation σ which is an important parameter of the model. Depending on the image noise, complexity of the object, and thickness of the boundaries, a proper value for σ can be chosen to control the blurring effect of the Gaussian function.

$$E_{\text{int}} = \frac{1}{2} \left(\alpha |p'(s)|^2 + \beta |p''(s)|^2 \right), \quad (3)$$

$$E_{\text{ext}} = \gamma |\nabla G_\sigma(x, y) * I(x, y)|^2. \quad (4)$$

Equations (3) and (4) represent the internal and external energies of the snake and the total snake energy is the sum of these two energies. Solving the classical snake model means minimizing the total snake energy, or in other words the sum of the integral of the two equations. In its original form, minimizing the energy function of the total snake energy gives rise to two independent Euler equations which

are explained in detail in the original reference (Kass et al. 1988). The classical snake model optimization approach, which uses the Euler equations for minimizing the snake energy, is a complex method because it requires calculating the higher-order derivatives of the contour.

The leading part of the PSO-Snake hybrid algorithm is its PSO component. PSO is a population-based evolutionary optimization algorithm. Population in the PSO is called a swarm and consists of a number of particles; each particle is a potential solution to the optimization problem depending on its location in the search space. Each particle, i , has a position, x_i and a speed, v_i , which are initialized with random values. Over a set of iterations, each particle's position on the search space is updated by revising its velocity according to its best experience, y_i , and its neighbors' experiences, \hat{y} . Particle position and its corresponding fitness value are stored as personal best experience and form the cognitive aspect of particle evolution. Another aspect of the particle position update is called the particle social behavior, which shows particles' influence on their neighbors. The PSO neighborhood can be defined with various topologies such as ring, star, von Neumann, and random. If all particles of the population are in the same neighborhood it is called global best (*gbest*) PSO, while if the neighborhood is restricted to a subset of the swarm it is called local best (*lbest*) PSO (used in this study). Figure 2 shows how the position of each PSO particle in the search space is updated under the influence of its velocity vectors. As it is shown in Figure 2, updating the particle velocity is also affected by the inertia velocity, i.e. the current velocity of the particle. Inertia velocity is important for controlling the particle velocity and preventing particle movement from radical changes. The following equations display the dynamics of the canonical PSO algorithm for updating particle velocity and position:

$$v_i(t + 1) = \omega(t)v_i(t) + c_1r_1(y_i(t) - x_i(t)) + c_2r_2(\hat{y}_i(t) - x_i(t)), \quad (5)$$

$$x_i(t + 1) = x_i(t) + v_i(t + 1), \quad (6)$$

where $x_i(t)$ and $v_i(t)$ are the position and the velocity of the i th particle at time t , $y_i(t)$ and $\hat{y}_i(t)$ denote the best positions discovered by the i th particle and its neighborhood up to the time t . The inertia weight, $\omega(t)$, controls the previous velocity impact. Usually the inertia weight is decreased dynamically during the run time, to balance between exploration in the early iterations and exploitation in the later iterations. Coefficients r_1 and r_2 are random numbers. Weights of cognitive

and social aspects of the algorithm are represented by acceleration factors c_1 and c_2 , respectively. As shown by Van den Bergh (2002) regulated values for inertia and acceleration weights can be used to achieve guaranteed convergence.

The PSO-Snake hybrid algorithm integrates the snake model mechanisms with PSO dynamics. While most swarm intelligence approaches in the literature that are used in conjunction with the snake model try to optimize the snake model equations, the PSO-Snake hybrid does not employ PSO algorithm only as a general problem solver to optimize snake energy minimization, but it also customizes the standard PSO to better solve this specific type of image processing problem. Early experiments on medical image segmentation (Shahamatnia & Ebadzadeh 2011) and sunspot tracking (Shahamatnia et al. 2012) reported promising results. It should be noted that running the standard snake model with the balloon force and the improved snake model with the Gradient Vector Flow forces (Xu & Prince 1998) on the CBP input images used in this paper did not yield any usable results. Due to the high noise in the image and weak CBP boundaries, the standard snake model initialized around a CBP could not converge to the CBP and evolved arbitrarily according to the sensitivity set by the balloon force. Testing the enhanced snake model improved the capture range of the image force, but still for the case of input images used in this paper it failed to be of practical use, since it requires meticulous parameter tuning for each individual CBP. A detailed discussion of the advantages of the PSO-Snake model over the traditional snake model is presented in (Shahamatnia & Ebadzadeh 2011).

In the Hybrid PSO-Snake model we use a population of particles where each particle is a snaxel of the contour. All particles together form the contour and hence the population is the final solution. As the algorithm runs, each particle updates its position and its velocity according to its personal best experience, local best experience, and according to the internal force of the snake and external force of the image. This gives the PSO-Snake dynamics a wider range of informative guides to update the particle position so that it converges to the region of interest.

In relation to the PSO algorithm (and the PSO-Snake for that matter), the term convergence refers to a stable condition where particles have found an optimum position in the search space (which might be a local or a global optimum). In general, the initial particle state is not at equilibrium, and convergence of the PSO is defined as the eventual settling of particles at the equilibrium (Trelea 2003). In PSO-Snake the equilibrium is reached when the snaxels latch to the boundaries of the object of interest, i.e. the CBPs within the initial contour.

PSO-Snake hybrid explores the search space according to the PSO trajectory disciplines. This eliminates the need to have a separate searching window around each particle as many swarm-based snake optimization algorithms do (Nebti & Meshoul 2009; Tseng et al. 2009; Horng et al. 2010). These methods consider a searching window around each particle and evaluate every position inside that window to determine the snaxels' next position. Since this local search is performed for each particle per iteration, it is a computationally expensive operation that is avoided in the PSO-Snake hybrid model. In contrast, the PSO trajectory disciplines do not require performing an exhaustive search for each particle move and hence alleviate the need for a local search window. According to the PSO kinematics, particles move toward the locations that they think will be promising by considering their own and their

neighbors' previous experiences. The PSO-Snake adopts the same concept and pushes snaxels to move toward promising locations according to some guiding terms. This is implemented via the PSO-Snake's velocity update equation:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 (pbest_i(t) - x_i(t)) + c_2 r_2 (lbest_i(t) - x_i(t)) + c_3 r_3 (\bar{x}(t) - x_i(t)) + c_4 (f.Image_i), \quad (7)$$

where $pbest_i(t)$ and $lbest_i(t)$ denote the best values experienced up to time t by particle i (personal best) and by its neighbors (local best), respectively. In the standard PSO, the fitness or the best experience of each particle is a location in the search space visited by the particle, for which the value of a specified objective function becomes minimum. In the PSO-Snake algorithm every particle constitutes a portion of the solution, hence the fitness of each particle should be calculated in regard to other particles. The fitness of the population corresponds to the snake energy and is calculated by Eq. (2). In the PSO-Snake algorithm the personal best experience of each particle is defined as the velocity of the particle in the best-experienced position (i.e. the objective function with the minimum value). Moreover, to update the $pbest$ the current fitness is not compared with previously calculated fitness for the particle's best experienced position, but the fitness is recalculated for that position in the new swarm. $\bar{x}(t)$ is the average of positions at time step t , approximating the center of mass of particles. This term pushes the snake to contract or expand with respect to the sign of its weighting factor, r_3 , speeds up the algorithm, and is particularly useful when the snake is stagnated and there is no other compelling force. $f.Image_i$ is the normalized image force corresponding to the external energy from snake model principles. For particle i , $f.Image_i$ gives the image force at the position specified by that particle. Image force can be any arbitrary function depending on the application, but generally external energies such as the image gradient and the gradient of a Gaussian functional are enough for satisfactory performance. It must also be noted that the image force does not vary by time and it is calculated only once for an image pixel. c_4 is the weighting factor to control the effect of image force. Inertia weight, ω , is taken to be a relatively small constant and r_1 , r_2 , and r_3 denote random numbers. In PSO-Snake, as well as traditional PSO algorithm, velocity values are influenced by a uniformly distributed random weight. That's because by its very nature, this algorithm is a stochastic search technique. Introducing some randomness in the search process can help to improve the algorithm speed in finding good results (Hoos & Stützle 2004) and several approximate global optimization algorithms such as simulated annealing, genetic algorithms, evolutionary strategies, PSO, and the bee algorithm employ mechanisms to inject randomness in their search process. Having a touch of randomness along with other forces guiding the particles helps the PSO-Snake algorithm to be more flexible in escaping from local minima and handling noise. Coefficients c_1 , c_2 , and c_3 are determined dynamically as a negative logarithmic function of $f.Image_i$ and coefficient c_4 is set as a constant coefficient. This ensures that if there is high image force, i.e. direct image information exists for guiding the particle evolution path, the image force will have a higher impact on the velocity update. Otherwise if the particle is out of the image force capture range, coefficients c_1 , c_2 , and c_3 will

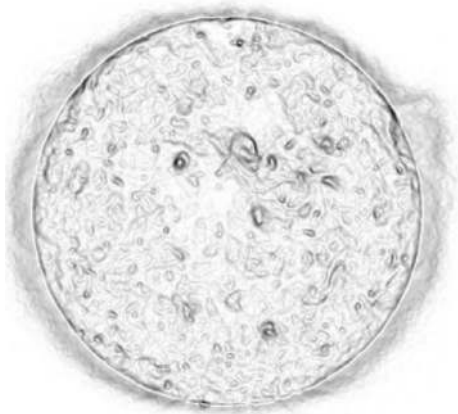


Fig. 3. Illustration of sample image force as a basis for calculating external snake energy.

take higher values and the particle evolution will be guided by other velocity update terms. The PSO-Snake hybrid algorithm uses l_{best} with ring structure and a radius of 3. The whole process can be summarized in the following steps:

Step 1. Initialization. If the input images need to be pre-processed it is done in this step. The algorithm requires all images in a tracking sequence to have the same resolution and image size. The RMS contrast measurement method explained in Peli (1990) can be used to check the consistency of image contrast in successive frames. Moreover for the coordinate logging system to work correctly, the orientation of the solar images should be corrected. The SDO-AIA images used in the current study are already preprocessed and do not require any further corrections.

Step 2. Initial Contour. The region of interest is chosen by the operator. This is the initial snake. For most cases a rough estimation of the initial contour is enough. This step is done only once when the coronal bright point appears.

Step 3. Internal parameters setup. The weight parameters for the PSO-Snake hybrid algorithm are initialized.

Step 4. Snake force calculation. The external force (image force) is calculated, once for every image. Depending on the application, noise, and complexity of the image, different image forces can be employed. In this work we use the gradient of the Gaussian function presented in Eq. (4) for calculating the image force. Figure 3 shows an illustration of sample image force calculated as a negative image.

Step 5. Calculation of social and cognitive parts. We update the p_{best} value (the best velocity the snaxel ever experienced) and the l_{best} value as the average of velocities of neighboring particles.

Step 6. Moving snaxels. Each snaxel's velocity is evaluated and then each snaxel's velocity and position are updated.

Step 7. Snake detection. The convergence of the snake contour to the coronal bright point outline is checked, i.e. choosing the snake with the lowest total energy calculated. If the results are not satisfactory, the algorithm goes back to step 5. The outcome of this step is the CBP contour for an image frame.

Step 8. Tracking CBPs. The same CBP in the next image is tracked by feeding the subsequent image frame to the system as input. The algorithm loops back to step 4 and passes the specifications of the detected CBP.

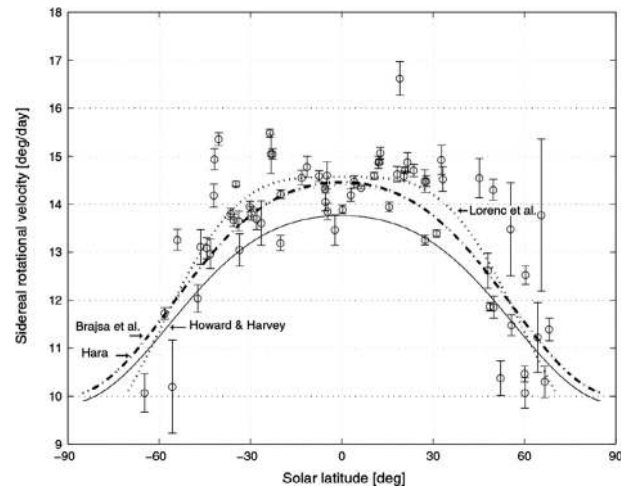


Fig. 4. Comparison of the derived values of the rotational speed. Points in the chart represent the measurements for the center of point-like structures with error bars showing the 95% confidence level intervals. The dotted curve shows the fit to the mean $\omega(b)$ values as a function of latitude b calculated by Eq. (8). Overplotted are the results of Howard & Harvey (1970) in solid line and Hara (2009) and Brajsa et al. (2004) both in the dashed-dotted curve because they are almost identical. Image courtesy of Lorenc et al. (2012).

Step 9. Stopping tracking. Tracking a CBP stops when it reaches the solar limb and disappears into the other side of the Sun, or when the CBP shrinks to a size smaller than a predefined threshold, according to the size and resolution of image.

3. Results and discussion

Our benchmark data are coronal images at 9.4 nm. This line is emitted by the Fe XVIII ion with a formation temperature of 7×10^6 K, that can dominate the emission in CBPs. This allows us to detect and track relatively bright structures. Images containing the CBPs with appropriate features such as proper cadence are selected from JPEG images taken between 14 September 2010 and 20 October 2010, downloaded from the SDO archive (<http://sdo.gsfc.nasa.gov/data/aiahmi/>). Altogether we have observed the motion of 24 CBP structures with small and simple shapes so that the data gathered for the evolution profile of the CBPs will be more reliable. Furthermore, it helps to avoid complex CBP evolution such as splitting CBPs. The CBPs used for tracking were chosen by an expert (coauthor) and the dataset used in this study is a subset of the dataset used by Lorenc et al. (2012). Tracking the 24 CBPs over their lifetime resulted in 1577 measurements. In a previous study (Lorenc et al. 2012), through a manual procedure, the CBP structures were observed directly on a PC monitor in an interactive session. Figure 4 shows latitudinal dependence of sidereal angular speed of coronal rotation obtained by the manual procedure in comparison with previous works. Further details can be found in Lorenc et al. (2012). In that paper, the position of each of the 4998 CBPs is determined by an expert operator manually on JPEG images and converted to heliographic coordinates. We employ the same method described in Lorenc et al. (2012) to compute the latitude b from the pixel coordinates of the tracked CBPs on the input images. In brief, in that method first the Cartesian coordinates of each CBP in solar image are converted to the spherical

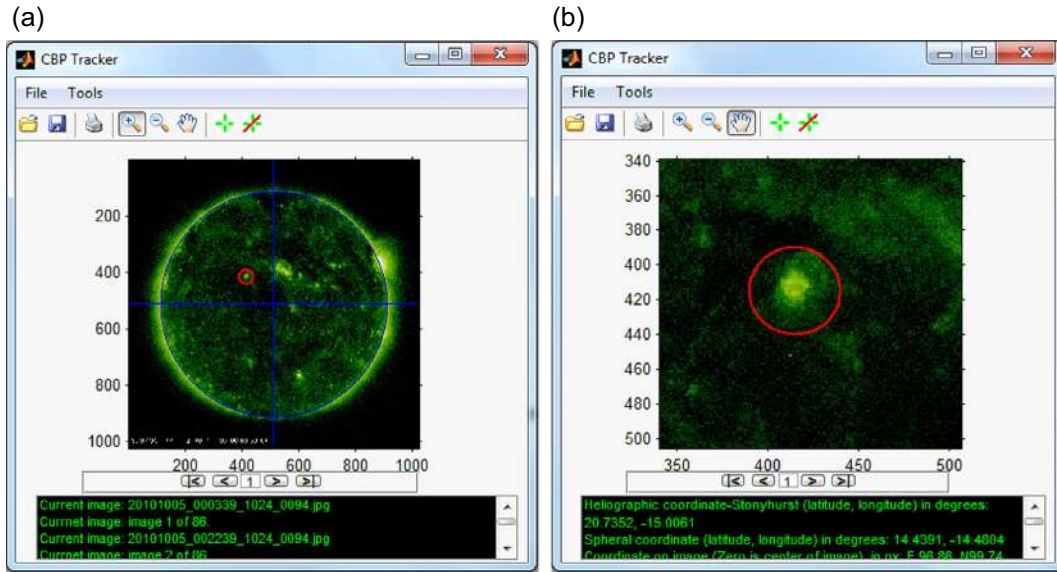


Fig. 5. A screenshot of the CBP Tracking system showing the Sun on 2010-10-05. (a) First snake (red circle) is initiated around the CBP we want to track, (b) zoomed-in view of the initial snake shows that the initial snake is a rough estimation of the location of the CBP. The algorithm will evolve the snake to find the exact CBP contour.

coordinate, knowing the radius of the Sun and center of the solar disk in the image. Then the spherical coordinate can be converted to the heliographic coordinate by adjusting to the tilt angle of the solar axis for the observation date of the CBP. This value is looked up from an ephemeris maintained by the BASS2000 archive (<http://bass2000.obspm.fr/ephem.php>).

In order to choose which CBP to track we use the initial location of the CBPs from the previous work. Initial snakes are defined as a circle encompassing the CBP. Then, we run our PSO-Snake hybrid algorithm to track those CBPs and calculate measurements. Input images are converted to grayscale color map with 256 gray levels per pixel and image force is calculated by a gradient of Gaussian functional with $\sigma = 3$ pixels. The value for σ is chosen as an empirical parameter. Input images have 1024×1024 resolution. For test purposes we have chosen the same CBPs for which we have the benchmark data available from the expert's manual CBP positioning. It should be noted that in the automated process, after choosing the CBP to be tracked (only once), the tracking process is automatic during the life span of that CBP.

Figure 5 shows a screenshot of the PSO-Snake hybrid algorithm tracking tool for a test image. The red circle shows the initial snake roughly locating the region of interest containing a CBP. Figure 6 shows how the initial snake is evolved under the PSO-Snake algorithm and the CBP boundary is detected. After the CBP is identified, its characteristics including the heliographic coordinates of its center of mass are calculated and stored. Then the next frame in the sequence is fed into the system. The detected CBP contour from the previous frame is used as a baseline to automatically track the CBP in the new frame. The temporal resolution of two successive images is calculated based on the image time stamp, then the new CBP initialization contour is started in the expected CBP location. Figure 7 shows a closer look at a tracked CBP along 37 frames. The results show the close matching between the automatic tracked positions and the manual CBP marking by an expert.

To compare the precision of the algorithm, we calculate several parameters proposed in Lorenc et al. (2012). In that

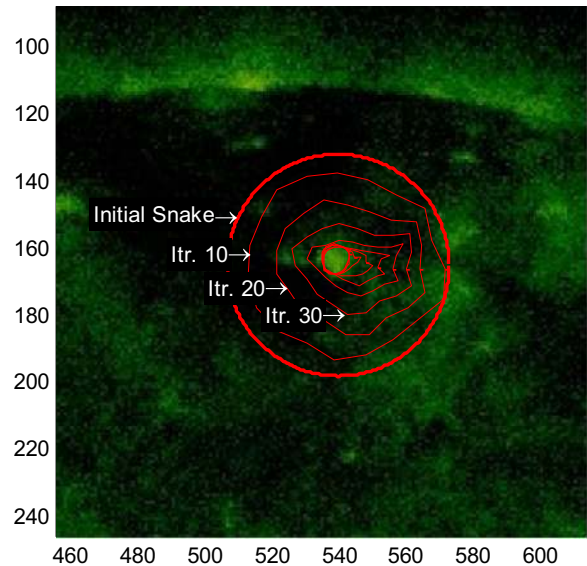


Fig. 6. Detection process of a selected CBP (2010-10-05). Initial contour around the region of interest is evolved to precisely delineate the CBP boundaries. The outer contour is the initial snake and the inner contours are the transitional contours of every 10 iterations of the PSO-Snake detection algorithm. In the final stage contours converge and do not change much through iterations.

paper, after an expert manually determined positions of CBPs on the solar images, the following measurements were calculated (reported in Table 1 of the referenced paper): angular rotation velocity denoted by ω and measurement error at 95% confidence level denoted by $\Delta\omega$. The Sun's surface rotational speed is calculated by Eq. (8) (Howard & Harvey 1970), where ω is the sidereal angular velocity of the rotation of the solar corona at latitude b and A , B , and C are the solar differential rotation parameters. Equation (8) is a common representation of the solar differential rotation (see references of Table 3). ω is expressed in [deg/day], b is expressed in [deg] and coefficients A , B , and C are in [deg/day]. Here we try to

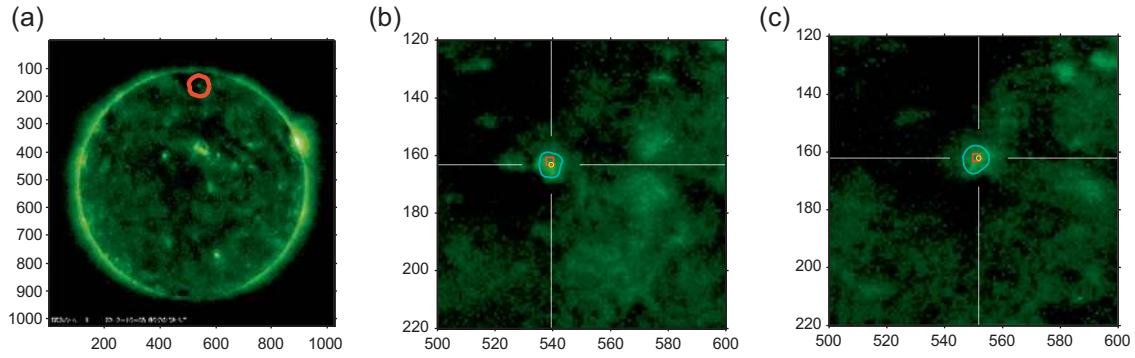


Fig. 7. (a) Red contour represents the initial snake on first image (the Sun at 2010-10-05), (b) and (c) cropped view of the tracking process of the selected CBP during time, showing the CBP in frame 1 and frame 37, respectively. The cyan contour is the boundary of the tracked CBP, the red square is the expert’s manual CBP positioning result, and the yellow circle is the PSO-Snake hybrid algorithm’s automated tracking result for CBP’s center of mass.

Table 1. Sample of results reported in Lorenc et al. (2012). *Structure* is the tracking ID given to each CBP as it can be seen in Figure 1, *n* is the number of frames for tracking the structure, *b* is the detected heliographic latitude of the structure, ω is the calculated angular rotation velocity, $\Delta\omega$ is the range of ω , and ω_E is the orbital angular rotation velocity of the Earth for the given time of observation.

Structure	<i>n</i>	<i>b</i> [deg]	ω [deg/day]	$\Delta\omega$ [deg/day]	ω_E [deg/day]
xy0510.01	86	66.7	10.29	±0.32	0.986
xy0510.03	86	20.5	14.58	±0.09	0.986
xy0510.04	86	-33.8	13.64	±0.20	0.986
xy0510.07	86	27.8	14.47	±0.11	0.986

Table 2. Results obtained by PSO-Snake hybrid algorithm for the same CBP structures of Table 1. *Structure* is the tracking ID of each CBP, *n* is the number of frames for tracking the structure, *b* is the detected heliographic latitude of the structure, ω is the calculated angular rotation velocity, $\Delta\omega$ is the range of ω , and ω_E is the orbital angular rotation velocity of the Earth for the given time of observation.

Structure	<i>n</i>	<i>b</i> [deg]	ω [deg/day]	$\Delta\omega$ [deg/day]	ω_E [deg/day]
xy0510.01	86	67.1	11.21	±0.64	0.986
xy0510.03	86	21	14.38	±0.30	0.986
xy0510.04	86	-32.2	13.80	±0.34	0.986
xy0510.07	86	28.1	15.11	±0.43	0.986

determine coefficients *A*, *B*, and *C* in order to best approximate the fitting of the heliographic latitude to the solar angular rotational velocity.

$$\omega(b) = A + B \sin^2 b + C \sin^4 b. \quad (8)$$

Figure 8 illustrates the difference between our calculated values and the benchmark values for the test CBPs. It should be noted that part of this deviation (10%–15%) is due to code implementation differences, which in precise calculations impose minute variations, called calculation error. Tables 1 and 2 show the results obtained with manual CBP tracking and results obtained by the PSO-Snake hybrid algorithm for some structures. In these tables, *structure* is the tracking ID given to each CBP as it can be seen in Figure 1, *n* is the number of frames the structure is tracked in, and *b* is the

detected heliographic latitude of the structure. ω_E is the orbital angular rotation velocity of the Earth, which can be looked up from solar almanacs. For the input images used in this study the almanac data is available only with a coarse temporal resolution, i.e. for the year 2010 the data is available for every 5 days. The exact ω_E for each CBP observation time is calculated by an interpolating technique.

In addition, the PSO-Snake results were assessed by an expert. Table 2 and Figure 9 show that the obtained results are very close to the result of manual CBP tracking. Computed angular rotation velocity is within ±0.2 [deg/day] of the benchmark data most of the time. It also worth mentioning that in several cases, where results were showing bigger differences, with further investigation we found out that PSO-Snake hybrid algorithm behaves consistently and the user-error was the main

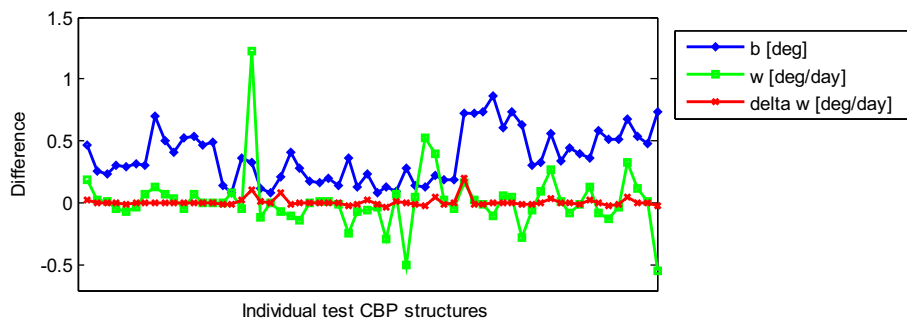


Fig. 8. Deviation of the obtained results for *b*, ω , $\Delta\omega$ on test data from the manual CBP marking of Lorenc et al. (2012). Every point in the horizontal axis represents a test CBP structure and its position in the vertical axis represents its error from the manual marking.

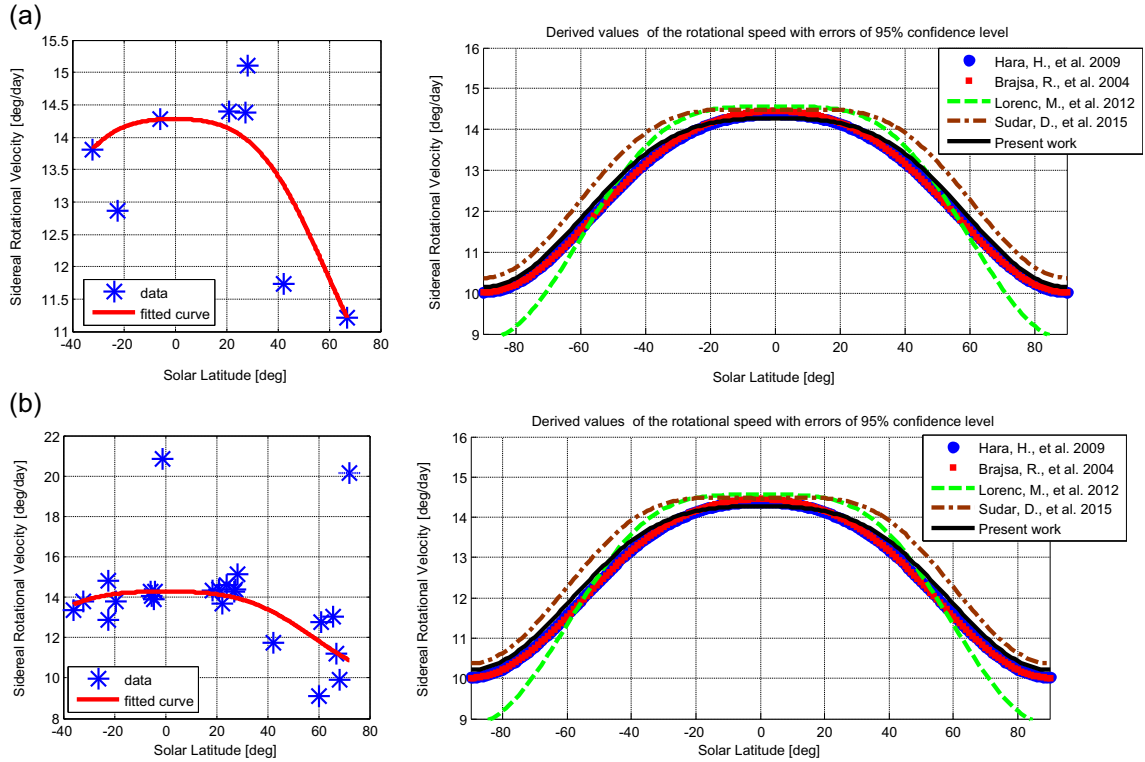


Fig. 9. Left: The result of Levenberg-Marquardt fitting algorithm (LMA) with least absolute residuals (LAR), on different number of tracked bright point structures. Right: Comparison of the curves based on Eq. (8) from different works. The black line represents this work, and it should be noticed that the similarity between the obtained results from this method with different numbers of tracked structures ((a) 650 CBP measurements and (b) 1577 CBP measurements) indicates the stability of this method.

Table 3. Comparison of results for the constant coefficients of the solar rotation profile (see Eq. (8)).

Work	A	B	C
Brajsa et al. 2004	14.45	-2.22	-2.22
Wöhl et al. 2010	14.50	-2.54	-0.77
Lorenc et al. 2012	14.56	0.00	-5.71
Brajsa et al. 2014	14.53	-0.59	-1.95
Sudar et al. 2015	14.47	0.60	-4.70
Present work – 650 measurements	14.28	-0.68	-3.44
Present work – 1577 measurements	14.28	-0.71	-3.35

cause of error. Table 3 compares the values obtained for the solar differential rotation constants A , B , and C of Eq. (8) in this method with results previously reported in the literature. As shown in Figure 9 left column, these coefficients are calculated by using a non-linear least squares curve-fitting function, with Levenberg-Marquardt algorithm (LMA) and least absolute residuals (LAR) estimation method and with no constraint applied on the coefficients. As part of the curve-fitting algorithm the confidence bounds on coefficients are set to be 95%. Figure 9 right column presents a visual comparison of Table 3 and illustrates how the calculated values for these coefficients determine the sidereal rotational velocity throughout all the solar latitudes. Early in our study, we had access to part of the manual database, and we did run our algorithm on 650 measurements. Later on, we got access to more data (with manual CBP markings for the comparison) and we ran the algorithm with more measurements (1577 measurements including the previous dataset). In Table 3, we decided to present the algorithm results with both the number of CBP

measurements to assess the scalability and stability of the method. It should be noted that the number of measurements from the comparison is not included in the table, because it is very different in terms of implementation, method used for the coefficient estimation, number of measurements used, representing date and solar cycle, dataset size, type of input images and channels, filtering methods, etc. This table only aims to compare the final results.

4. Conclusions and future works

In this paper the PSO-Snake hybrid algorithm has been used to address a real-world problem from the solar physics domain. The PSO-Snake hybrid model inherits a particle navigation scheme from PSO and modifies it to include snake model algorithm concepts. Combining PSO dynamics with snake model kinematics enables us to successfully overcome active contour difficulties, while preserving the simplicity of PSO. A detailed discussion of the advantages of PSO-Snake model over traditional snake model is presented in Shahamatnia & Ebadzadeh (2011). Comparing Eq. (5), the standard PSO velocity update equation, with Eq. (7) of the PSO-Snake algorithm, shows that two new terms have been added to the PSO-Snake algorithm, namely the $(\bar{x}(t) - x_i(t))$ that refers to the average of positions at time t and $(f.Image_i)$, which is the normalized image force corresponding to external energy from snake model principles. By adding two new terms to the PSO velocity update equations, the PSO-Snake model still can evolve even if some of the components are missing or misleading.

Other aspects of the PSO were also changed in the PSO-Snake hybrid model. For example, the population in the

PSO-Snake represents the positions of snaxels and hence, the whole population forms one viable solution to the problem. The PSO-Snake model is similar to the usual snake formulations in the definition of snaxels and external forces. However, unlike the usual snake formulation, which optimizes the snake by solving the Euler-Lagrange equations or represents it as a level-set formulation, PSO-Snake inherits PSO kinematics by moving the snaxels in the search space toward the global optimum (i.e. the final contour). This is fundamentally different from the snake's elastic boundary constraints because the PSO particle evolving mechanisms are based on particles' experience in the search space, while the usual snake formulations emphasize regulating the shape and physical characteristics of the contour. Elastic boundary constraints can be set to restrain the movement of a snaxel in a way that snaxels would demonstrate a desired physical characteristic. In contrast, PSO particles move in the search space irrespective of such constraints, i.e. they evaluate the position that they are visiting (underlying pixel) and tend to move to the promising areas by sharing information with their neighbors. In other words, while the elastic boundary constraint emphasizes the shape of the contour, the PSO-Snake emphasizes the search direction.

This algorithm uses velocity vectors to guide particles in the promising direction based on the information obtained from the search space that the particle itself or its neighbors have visited before. This is all implemented with the notion of consolidated velocity vectors, including the inertia of particles. PSO-Snake model can be used for different applications in image processing for object detection, image segmentation and tracking. Since the particle/snaxels have embedded velocity information, which can adapt to the object movement over time, this method is able to naturally incorporate and exploit that information.

By tracking the CBPs over time, the angular rotational velocity of the Sun is calculated. Lorenc et al. (2012) used a manual method to track and mark the location of each individual CBP in a dataset, however, the method is laborious and with a large number of images becomes unworkable for practical reasons. In this work we devised an adaptation of the PSO-Snake to automate the manual method of Lorenc et al. (2012) in tracking CBPs. Further, the human-defined dataset of Lorenc et al. (2012) was used as the ground truth for validating our algorithm. The obtained CBP tracking results were used to estimate the sidereal rotational angular velocity of the rotation of the solar corona. Further, we compared our results for the angular rotational velocity of the Sun with those reported in the literature. The results obtained in this study by running the PSO-Snake algorithm to track CBPs, on a sample dataset, showed good conformity with previous works. Despite the limited number of test cases used in this study, the proposed algorithm proved to be stable and gives consistent results. Moreover, we expect the parameter errors to reduce by running the algorithm on larger datasets. Higher number of structures would allow easier detection and removal of the outlier data and better sampling of data in all latitudes, thus reducing the parameter error in the curve-fitting function. Outliers could be due to human error, wrong database entry, calculation error, or eventually the algorithm failure. Detection and removal of these values will help improve the quality of results for the sidereal angular rotational velocity calculation. With larger datasets and sampling of data from all latitudes, the detection and removal of outlier data would be easier. We expect that an automatic outlier detection algorithm based on Peirce's criterion would be suitable as future work.

In summary, this paper's objective was to introduce the PSO-Snake algorithm to the solar physics community and report early results on CBP tracking problems. The algorithm showed the feasibility of using it for solar image analysis by addressing a current solar physics question. At the current state, the algorithm is suitable for tracking solar features, pre-marked by a user or indicated as an outcome from another system. As future work, authors are extending the PSO-Snake algorithm to make it fully automated, including initial detection of the CBPs. This will allow to analyze CBPs automatically, for extensive number of solar images and compare with the results of McIntosh & Gurman (2005), obtained by analyzing nine years of SOHO EUV images. Also it should be noticed that JPEG images were used in this study for their ease of use and as a proof of concept for the algorithm, but we will extend this work to use science-grade FITS images, for improved quality and precision.

Acknowledgements. We thank the SDO (NASA) and AIA science team for the provided observational material. This work was partially supported by Grants SFRH/BPD/44018/2008 (I.D.) and SFRH/BD/62249/2009 (E.S.) from Fundação para a Ciência e a Tecnologia (FCT), MCTES, Portugal, and by FCT Strategic Program UID/EEA/00066/203 of UNINOVA, CTS. The editor thanks three anonymous referees for their assistance in evaluating this paper.

References

- Agrawal, J., and S. Agrawal. Acceleration based Particle Swarm Optimization (APSO) for RNA secondary structure prediction, *Progress in Systems Engineering*, Volume 330 of the series Advances in Intelligent Systems and Computing, Springer International Publishing, Switzerland,, 741–746, 2015, DOI: [10.1007/978-3-319-08422-0_106](https://doi.org/10.1007/978-3-319-08422-0_106).
- Amini, A., S. Tehrani, and T.E. Weymouth. Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints, Second International Conference on Computer Vision, Tampa, FL, 95–99, 1988, DOI: [10.1109/CCV.1988.589976](https://doi.org/10.1109/CCV.1988.589976).
- Asl, M.A., and S.A. Seyedin. Active Contour Optimization Using Particle Swarm Optimizer, 2nd International Conference on Information & Communication Technologies, Damascus, 1, 522–523, 2006, DOI: [10.1109/ICTTA.2006.1684608](https://doi.org/10.1109/ICTTA.2006.1684608).
- Ballerini, L. Genetic snakes for medical images segmentation. *Evolutionary Image Analysis, Signal Processing and Telecommunications*, **1596**, 59–73, 1999, DOI: [10.1007/10704703_5](https://doi.org/10.1007/10704703_5).
- Ballerini, L., and L. Bocchi. Multiple genetic snakes for bone segmentation, Applications of Evolutionary Computing, Essex, UK, 346–356, 2003, DOI: [10.1007/3-540-36605-9_32](https://doi.org/10.1007/3-540-36605-9_32).
- Brajša, R., H. Wöhl, V. Ruždjak, F. Clette, and J.F. Hochedez. Solar differential rotation determined by tracing coronal bright points in SOHO-EIT images I. Interactive and automatic methods of data reduction. *A&A*, **374**, 309–315, 2001, DOI: [10.1051/0004-6361:20010694](https://doi.org/10.1051/0004-6361:20010694).
- Brajša, R., D. Sudar, I. Skokić, and S.H. Saar. Preliminary results on the solar rotation determined tracing SDO/AIA coronal bright points. *Cent. Eur. Astrophys. Bull.*, **38**, 105–110, 2014.
- Brajša, R., H. Wöhl, B. Vršnak, V. Ruždjak, F. Clette, J.F. Hochedez, and D. Roša. Height correction in the measurement of solar differential rotation determined by coronal bright points. *A&A*, **414**, 707–715, 2004, DOI: [10.1051/0004-6361:20034082](https://doi.org/10.1051/0004-6361:20034082).
- Bresson, X., S. Esedoğlu, P. Vanderghenst, J.P. Thiran, and S. Osher. Fast global minimization of the active contour/snake model. *J. Math. Imaging Vis.*, **28**, 151–167, 2007, DOI: [10.1007/s10851-007-0002-0](https://doi.org/10.1007/s10851-007-0002-0).
- Brown, D.S., C.E. Parnell, E.E. Deluca, L. Golub, and R.A. McMullen. The magnetic structure of a coronal X-ray bright point. *Sol. Phys.*, **201**, 305–321, 2001, DOI: [10.1023/A:1017907406350](https://doi.org/10.1023/A:1017907406350).

- Caselles, V., R. Kimmel, and G. Sapiro. Geodesic active contours. *Int. J. Comput. Vision*, **22**, 61–79, 1997, DOI: [10.1023/A:1007979827043](https://doi.org/10.1023/A:1007979827043).
- Cohen, L.D., and I. Cohen. Finite-element methods for active contour models and balloons for 2-D and 3-D images. *IEEE Trans. Pattern Anal. Mach. Intell.*, **15**, 1131–1147, 1993, DOI: [10.1109/34.244675](https://doi.org/10.1109/34.244675).
- Davatzikos, C., and J.L. Prince. Convexity analysis of active contour models, Proc. Conf. Info. Sci. Sys., Princeton, NJ, 581–587, 1994.
- Eberhart, R.C., and Y. Shi. Tracking and optimizing dynamic systems with particle swarms, Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2001), Seoul, 94–97, 2001, DOI: [10.1109/CEC.2001.934376](https://doi.org/10.1109/CEC.2001.934376).
- Eberhart, R.C., and X. Hu. Human tremor analysis using particle swarm optimization, Proceedings of the 1999 Congress on Evolutionary Computation-CEC99, Washington, DC, 3, 1999, DOI: [10.1109/CEC.1999.785508](https://doi.org/10.1109/CEC.1999.785508).
- Ethni, S.A., B. Zahawi, D. Giaouris, and P.P. Acarnley. Comparison of particle swarm and simulated annealing algorithms for induction motor fault identification, IEEE International Conference on Industrial Informatics (INDIN), Cardiff, Wales, 470–474, 2009, DOI: [10.1109/INDIN.2009.5195849](https://doi.org/10.1109/INDIN.2009.5195849).
- Habbal, S.R., and G.L. Withbroe. Spatial and temporal variations of EUV coronal bright points. *Sol. Phys.*, **69**, 77–97, 1981, DOI: [10.1007/BF00151257](https://doi.org/10.1007/BF00151257).
- Habib, S.J., and B.S. Al-kazemi. Comparative study between the internal behavior of GA and PSO through problem-specific distance functions, IEEE Congress on Evolutionary Computation, Edinburgh, Scotland, UK, 2005, DOI: [10.1109/CEC.2005.1554966](https://doi.org/10.1109/CEC.2005.1554966).
- Hanasoge, S., M.S. Miesch, M. Roth, J. Schou, and M.J. Thompson. Solar dynamics, rotation, convection and overshoot. *Space Sci. Rev.*, **196**, 79–99, 2015, DOI: [10.1007/s11214-015-0144-0](https://doi.org/10.1007/s11214-015-0144-0).
- Hara, H. Differential rotation rate of X-ray bright points and source region of their magnetic fields. *Astrophys. J.*, **697**, 980, 2009, DOI: [10.1088/0004-637X/697/2/980](https://doi.org/10.1088/0004-637X/697/2/980).
- Hassan, R., B.E. Cohanin, O.L. de Weck, and G. Venter. A comparison of particle swarm optimization and the genetic algorithm, Proceedings of the 1st AIAA Multidisciplinary Design Optimization Specialist Conference, Austin, Texas, American Institute of Aeronautics and Astronautics, 2005, DOI: [10.2514/6.2005-1897](https://doi.org/10.2514/6.2005-1897).
- He, N., P. Zhang, and K. Lu. A geometric active contours model for multiple objects segmentation, *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues*, Springer-Verlag, Berlin Heidelberg, 1141–1148, 2008, DOI: [10.1007/978-3-540-87442-3_141](https://doi.org/10.1007/978-3-540-87442-3_141).
- Holden, N., and A.A. Freitas. Hybrid particle swarm/ant colony algorithm for the classification of hierarchical biological data, Proc. IEEE Int. Symp. Swarm Intelligence, Pasadena, CA, USA, 100–107, 2005, DOI: [10.1109/SIS.2005.1501608](https://doi.org/10.1109/SIS.2005.1501608).
- Hoos, H.H., and T. Stützle. *Stochastic Local Search: Foundations & Applications*, Elsevier, San Francisco, CA, 2004.
- Hornig, M.H., R.J. Liou, and J. Wu. Parametric active contour model by using the honey bee mating optimization. *Expert Syst. Appl.*, **37**, 7015–7025, 2010, DOI: [10.1016/j.eswa.2010.03.017](https://doi.org/10.1016/j.eswa.2010.03.017).
- Howard, R. Solar rotation. *Ann. Rev. Astron. Astrophys.*, **22**, 131–155, 1984, DOI: [10.1146/annurev.aa.22.090184.001023](https://doi.org/10.1146/annurev.aa.22.090184.001023).
- Howard, R., and J. Harvey. Spectroscopic determinations of solar rotation. *Sol. Phys.*, **12**, 23–51, 1970, DOI: [10.1007/BF02276562](https://doi.org/10.1007/BF02276562).
- Howard, R., and B.J. LaBonte. The Sun is observed to be a torsional oscillator with a period of 11 years. *Astrophys. J.*, **239**, L33–36, 1980.
- Hu, X., Y. Shi, and R. Eberhart. Recent advances in particle swarm, Proceedings of the 2004 Congress on Evolutionary Computation, Portland, OR, USA, 1, 2004, DOI: [10.1109/CEC.2004.1330842](https://doi.org/10.1109/CEC.2004.1330842).
- Karlsson, A., K. Stråhlén, and A. Heyden. A fast snake segmentation method applied to histopathological sections, Energy Minimization Methods in Computer Vision and Pattern Recognition, Lisbon, Portugal, 261–274, 2003, DOI: [10.1007/978-3-540-45063-4_17](https://doi.org/10.1007/978-3-540-45063-4_17).
- Kass, M., A. Witkin, and D. Terzopoulos. Snakes: active contour models. *Int. J. Comput. Vision*, **1**, 321–331, 1988, DOI: [10.1007/BF00133570](https://doi.org/10.1007/BF00133570).
- Kennedy, J., and R. Eberhart. Particle swarm optimization, IEEE International Conference on Neural Networks, Perth, WA, 4, 1942–48, 1995, DOI: [10.1109/ICNN.1995.488968](https://doi.org/10.1109/ICNN.1995.488968).
- Lam, K.M., and H. Yan. Fast Greedy algorithm for active contours. *Electron. Lett.*, **30**, 21–23, 1994, DOI: [10.1049/el:19940040](https://doi.org/10.1049/el:19940040).
- Lemen, J.R., D.J. Akin, P.F. Boerner, C. Chou, J.F. Drake, et al. The Atmospheric Imaging Assembly (AIA) on the Solar Dynamics Observatory (SDO). *Sol. Phys.*, **275**, 17–40, 2012, DOI: [10.1007/s11207-011-9776-8](https://doi.org/10.1007/s11207-011-9776-8).
- Leroy, B., I.L. Herlin, and L.D. Cohen. Multi-resolution algorithms for active contour models. *ICAOS'96*, **219**, 58–65, 1996, DOI: [10.1007/3-540-76076-8_117](https://doi.org/10.1007/3-540-76076-8_117).
- Li, B., and S.T. Acton. Active contour external force using vector field convolution for image segmentation. *IEEE Trans. Image Process.*, **16**, 2096–2106, 2007, DOI: [10.1109/TIP.2007.899601](https://doi.org/10.1109/TIP.2007.899601).
- Li, R., Y. Guo, Y. Xing, and M. Li. A novel multi-swarm particle swarm optimization algorithm applied in active contour model, WRI Global Congress on Intelligent Systems, GCIS '09, 139–143, 2009, DOI: [10.1109/GCIS.2009.57](https://doi.org/10.1109/GCIS.2009.57).
- Lorenc, M., M. Rybanský, and I. Dorotovič. On rotation of the solar corona. *Sol. Phys.*, **281**, 611–619, 2012, DOI: [10.1007/s11207-012-0105-7](https://doi.org/10.1007/s11207-012-0105-7).
- Marinakis, Y., M. Marinaki, and G. Dounias. A hybrid particle swarm optimization algorithm for the vehicle routing problem. *Eng. Appl. Artif. Intell.*, **23**, 463–472, 2010, DOI: [10.1016/j.engappai.2010.02.002](https://doi.org/10.1016/j.engappai.2010.02.002).
- McInerney, T., and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Med. Image Anal.*, **1**, 91–108, 1996, DOI: [10.1016/S1361-8415\(96\)80007-7](https://doi.org/10.1016/S1361-8415(96)80007-7).
- McIntosh, S.W., and J.B. Gurman. Nine years of EUV bright points. *Sol. Phys.*, **228**, 285–299, 2005, DOI: [10.1007/s11207-005-4725-z](https://doi.org/10.1007/s11207-005-4725-z).
- Mun, K.J., H.T. Kang, H.S. Lee, Y.S. Yoon, C.M. Lee, and J.H. Park. Active contour model based object contour detection using genetic algorithm with wavelet based image preprocessing. *Int. J. Control Autom. Syst.*, **2**, 100–106, 2004.
- Nefti, S., and S. Meshoul. Predator prey optimization for snake-based contour detection. *International Journal of Intelligent Computing and Cybernetics*, **2**, 228–242, 2009, DOI: [10.1108/17563780910959884](https://doi.org/10.1108/17563780910959884).
- Niu, X. A geometric active contour model for highway extraction, Proceedings of ASPRS 2006 Annual Conference, Reno, Nevada, 2006.
- Panda, S., and N.P. Padhy. Comparison of particle swarm optimization and genetic algorithm for FACTS-based controller design. *Applied Soft Computing*, **8**, 1418–1427, 2008, DOI: [10.1016/j.asoc.2007.10.009](https://doi.org/10.1016/j.asoc.2007.10.009).
- Paragios, N., and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, **22**, 266–280, 2000, DOI: [10.1109/34.841758](https://doi.org/10.1109/34.841758).
- Park, H.W., T. Schoepflin, and Y. Kim. Active contour model with gradient directional information: directional snake. *IEEE Trans. Circuits Syst. Video Technol.*, **11**, 252–256, 2001, DOI: [10.1109/76.905991](https://doi.org/10.1109/76.905991).
- Peli, E. Contrast in complex images. *J. Opt. Soc. Am.*, **A7**, 2032–2040, 1990, DOI: [10.1364/JOSAA.7.002032](https://doi.org/10.1364/JOSAA.7.002032).
- Pesnell, W.D., B.J. Thompson, and P.C. Chamberlin. The Solar Dynamics Observatory (SDO). *Sol. Phys.*, **275**, 3–15, 2012, DOI: [10.1007/s11207-011-9841-3](https://doi.org/10.1007/s11207-011-9841-3).
- Poli, R. Analysis of the publications on the applications of particle swarm optimisation. *JAEA*, **2008**, 1–10, 2008, DOI: [10.1155/2008/685175](https://doi.org/10.1155/2008/685175).

- Prince, J.L. Gradient vector flow: a new external force for snakes. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, 2, 66–71, 1997, DOI: [10.1109/CVPR.1997.609299](https://doi.org/10.1109/CVPR.1997.609299).
- Ram, G., D. Mandal, R. Kar, and S.P. Ghosal. Synthesis of time modulated linear antenna arrays using particle swarm optimization, IEEE Region 10 Conference, TENCON'14, Bangkok, 1–4, 2014, DOI: [10.1109/TENCON.2014.7022315](https://doi.org/10.1109/TENCON.2014.7022315).
- Robinson, J., S. Sinton, and Y. Rahmat-Samii. Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna, IEEE Antennas and Propagation Society International Symposium, 314–317, 2002, DOI: [10.1109/APS.2002.1016311](https://doi.org/10.1109/APS.2002.1016311).
- Scherrer, P.H., J. Schou, R.I. Bush, A.G. Kosovichev, R.S. Bogart, et al. The Helioseismic and Magnetic Imager (HMI) investigation for the Solar Dynamics Observatory (SDO), *The Solar Dynamics Observatory*, Springer, US, 207–227, 2011, DOI: [10.1007/978-1-4614-3673-7_10](https://doi.org/10.1007/978-1-4614-3673-7_10).
- Shahamatnia, E., I. Dorotovič, R.A. Ribeiro, and J.M. Fonseca. Towards an automatic sunspot tracking: swarm intelligence and snake model hybrid. *Acta Futura*, 5, 153–61, 2012, DOI: [10.2420/AF05.2012.153](https://doi.org/10.2420/AF05.2012.153).
- Shahamatnia, E., and M.M. Ebadzadeh. Application of particle swarm optimization and snake model hybrid on medical imaging, IEEE Third International Workshop on Computational Intelligence In Medical Imaging, Paris, France, 2011, DOI: [10.1109/CIMI.2011.5952043](https://doi.org/10.1109/CIMI.2011.5952043).
- Sudar, D., I. Skokic, R. Brajša, and H. Saar. Steps toward a high precision solar rotation profile: results from SDO/AIA coronal bright point data. *A&A*, 575, A63, 2015, DOI: [10.1051/0004-6361/201424929](https://doi.org/10.1051/0004-6361/201424929).
- Thompson, M.J., J. Toomre, E.R. Anderson, H.M. Antia, G. Berthomieu, et al. Differential rotation and dynamics of the solar interior. *Science*, 272, 1300–1305, 1996, DOI: [10.1126/science.272.5266.1300](https://doi.org/10.1126/science.272.5266.1300).
- Trelea, I.C. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Inform. Process. Lett.*, 85, 317–325, 2003, DOI: [10.1016/S0020-0190\(02\)00447-7](https://doi.org/10.1016/S0020-0190(02)00447-7).
- Tseng, C.C., J. Hsieh, and J. Jeng. Active contour model via multi-population particle swarm optimization. *Expert Syst. Appl.*, 36, 5348–5352, 2009, DOI: [10.1016/j.eswa.2008.06.114](https://doi.org/10.1016/j.eswa.2008.06.114).
- Van den Bergh, F. An analysis of particle swarm optimizers, *Ph.D. dissertation*, University of Pretoria, 2002.
- Wildenauer, H., P. Blauensteiner, A. Hanbury, and M. Kampel. Motion detection using an improved colour model, *Advances in Visual Computing*, Springer, Berlin Heidelberg, 607–616, 2006, DOI: [10.1007/11919629_61](https://doi.org/10.1007/11919629_61).
- Wöhl, H., R. Brajša, A. Hanslmeier, and S.F. Gissot. A precise measurement of the solar differential rotation by tracing small bright coronal structures in SOHO-EIT images. *A&A*, 520, A29, 2010, DOI: [10.1051/0004-6361/200913081](https://doi.org/10.1051/0004-6361/200913081).
- Xu, C., and J.L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Trans. Image Process.*, Pacific Grove, CA, USA, 7, 359–369, 1998, DOI: [10.1109/83.661186](https://doi.org/10.1109/83.661186).
- Xu, C., A. Yezzi, and J.L. Prince. On the relationship between parametric and geometric active contours, Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers, 1, 483–489, 2000, DOI: [10.1109/ACSSC.2000.911003](https://doi.org/10.1109/ACSSC.2000.911003).
- Yang, X.S., Editor. *Recent Advances in Swarm Intelligence and Evolutionary Computation*, 1st ed., Vol. 585, Springer International Publishing, Switzerland, 2015, DOI: [10.1007/978-3-319-13826-8](https://doi.org/10.1007/978-3-319-13826-8).
- Yang, F., C. Zhang, and T. Sun. Comparison of particle swarm optimization and genetic algorithm for HMM training, 19th International Conference on Pattern Recognition, Tampa, FL, 2008, DOI: [10.1109/ICPR.2008.4761282](https://doi.org/10.1109/ICPR.2008.4761282).
- Zeng, D., and Z. Zhou. Invariant topology snakes driven by particle swarm optimizer, 3rd International Conference on Innovative Computing Information and Control, Dalian, Liaoning, 38–38, 2008, DOI: [10.1109/ICICIC.2008.332](https://doi.org/10.1109/ICICIC.2008.332).

Cite this article as: Shahamatnia E, Dorotovič I, Fonseca JM & Ribeiro RA. An evolutionary computation based algorithm for calculating solar differential rotation by automatic tracking of coronal bright points. *J. Space Weather Space Clim.*, 6, A16, 2016, DOI: [10.1051/swsc/2016010](https://doi.org/10.1051/swsc/2016010).