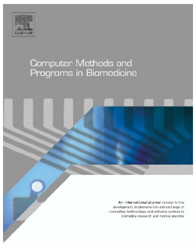
journal homepage: www.intl.elsevierhealth.com/journals/cmpb



UltraTrack: Software for semi-automated tracking of muscle fascicles in sequences of B-mode ultrasound images

Dominic James Farris[*], *Glen A. Lichtwark*

School of Human Movement & Nutrition Sciences, Level 5, Building 26B, Blair Drive, The University of Queensland, Brisbane, QLD 4072, Australia

ARTICLE INFO

Article history:
Received 9 November 2015
Received in revised form
14 January 2016
Accepted 24 February 2016

Keywords:
Fascicle
Tracking
Ultrasound
Automated
Affine optic flow
Matlab

ABSTRACT

Background: Dynamic measurements of human muscle fascicle length from sequences of B-mode ultrasound images have become increasingly prevalent in biomedical research. Manual digitisation of these images is time consuming and algorithms for automating the process have been developed. Here we present a freely available software implementation of a previously validated algorithm for semi-automated tracking of muscle fascicle length in dynamic ultrasound image recordings, “UltraTrack”.

Methods: UltraTrack implements an affine extension to an optic flow algorithm to track movement of the muscle fascicle end-points throughout dynamically recorded sequences of images. The underlying algorithm has been previously described and its reliability tested, but here we present the software implementation with features for: tracking multiple fascicles in multiple muscles simultaneously; correcting temporal drift in measurements; manually adjusting tracking results; saving and re-loading of tracking results and loading a range of file formats.

Results: Two example runs of the software are presented detailing the tracking of fascicles from several lower limb muscles during a squatting and walking activity.

Conclusion: We have presented a software implementation of a validated fascicle-tracking algorithm and made the source code and standalone versions freely available for download.

1. Introduction

The in vivo measurement of human skeletal muscle function has long been of interest to scientists and engineers from a variety of backgrounds including: physiology, anatomy, evolutionary biology, ergonomics, biomedical engineering and human movement science. Of fundamental importance in this area is the measurement of muscle architecture and how it dynamically changes with muscle contraction. A key feature of muscle architecture is muscle contractile element length and many important research questions can be answered with knowledge of how contractile elements change length during muscle contraction. The basic contractile element of muscle is the sarcomere and many sarcomeres in series make up muscle fibres, which in turn are bundled into fascicles. Sarcomeres are microscopic in size and so it is common to measure fibre or fascicle length changes as a proxy.

[*] *Corresponding author*. Tel.: +61 07 3365 6097.
E-mail addresses: d.farris@uq.edu.au (D.J. Farris), g.lichtwark@uq.edu.au (G.A. Lichtwark).

http://dx.doi.org/10.1016/j.cmpb.2016.02.016

In the last twenty years, B-mode ultrasound has risen in popularity as a means of imaging skeletal muscle in static and dynamic conditions. Cross-sectional ultrasound images of skeletal muscle allow identification of muscle fascicles [1]. By sampling images at a suitable rate, the length changes of muscle fascicles during dynamic contractions have been characterised through digitisation of image sequences [2,3]. Initially, the digitisation process was manual, making it extremely time consuming and somewhat subjective. More recently, image processing algorithms to partially or fully automate the digitisation process have been developed [4–6]. These employ a variety of methods including Kanade–Lucas–Tomasi feature tracking and affine transformation extensions to optic flow. The validity and reliability of the affine optic flow algorithm of Cronin et al. [4] was published some time ago [4,7]. The algorithm and the software implementation have since been substantially expanded and in this paper we present the refined algorithm and freely available software package, UltraTrack version 4.1, the source code for which is written in Matlab (The MathWorks Inc., USA). A new 'key-frame correction' algorithm has been added to help remove temporal drift (accumulation of error with time) in the tracking that had previously limited the length of image sequences that could be tracked. A manual correction is also added as an alternative for making adjustments to tracking results. New features of the software include: compatibility with more image file formats, faster implementation of the affine flow algorithm, tracking of multiple fascicles and regions of interest, image auto-cropping function, saving and re-loading of tracking data and a choice of moving or fixed region of interest for the affine flow calculations.

2. Computational methods

2.1. *Overview*

The software requires an image sequence as input (see Section 3.1 for details) and the user must manually define region(s) of interest and the representative fascicle(s) that they wish to track (see Section 3.3 for details) in the image frame from which they wish to start tracking. Once region(s) and fascicle(s) are defined, the user can process the image sequence, which implements an algorithm based on the affine optic flow model described in Section 2.2 to track fascicle length in the subsequent frames of the image sequence. The algorithm steps through the image sequence one frame at a time, computing the optic flow between consecutive images and applying the affine transformation to calculate the new position and length of the fascicle(s). Because this is an iterative process, any tracking errors in individual frames may accumulate over time and be compounded, causing the fascicle position and length to 'drift' as the algorithm progresses through image sequences. To correct for this, we have developed the '*key-frame correction*' method that is described in Section 2.3.

2.2. *The affine optic flow model*

The main algorithm in the software implements an affine optic flow model to compute the optic flow and the affine transformation between consecutive images in the sequence. Once the model parameters have been computed for a pair of images, the affine transformation matrix can be used to determine the new Cartesian coordinates of any point defined in the first image, in the second image. This process is applied to compute the displacement of the fascicle endpoints from one image to the next. The model and algorithm have been described and their use validated previously [4,7], but for completeness we will also describe it here.

The model is based upon an affine extension of the well-established Lucas–Kanade method [8]. The affine optic flow model has six parameters: v_{xt} – optic flow at the image origin (top left corner) in the *x*-direction; v_{yt} – optic flow at the origin (top left corner) in the *y*-direction; *d* – rate of dilation; *r* – rate of rotation; s_1 – shear along the main image axis; s_2 – shear along the diagonal image axis. These parameters can be used to estimate the flow vector (change in position *vx*, *vy*) at specific points in the image (*x*, *y*) by applying the following first order model:

$$(v_x, v_y) = \begin{bmatrix} x & t & 1 \end{bmatrix} \times \begin{bmatrix} d+s_1 & s_2+r \\ s_2-r & d-s_1 \\ v_{xt} & v_{yt} \end{bmatrix} \tag{1}$$

The Matlab algorithm used here (contributed by Dr. David Young and available from Matlab Central – see acknowledgments) implements a least squares fit of the above parameters to estimates of the spatial and temporal grey-level gradients on a rectilinear grid within the defined region of interest. To calculate the spatial and temporal gradients, the images are first smoothed in both the x and y components by convoluting the image with a 1D Gaussian mask. In the implementation used here, the Matlab function "fspecial" is used to create the mask with a width of 2.6 times the parameter "sigma," which is the standard deviation of the distribution. To calculate the spatial gradients, the symmetric local difference of the average of the two smoothed images is calculated by two-dimensional convolution in the *x* and *y* directions using a simple linear gradient function. The temporal gradients are calculated as the difference between the two smoothed images. The affine flow parameters are then solved by a least squares fit of the parameters to achieve the given spatial and temporal gradients. The x and y grids along with the spatial and temporal gradients are then resampled (3 pixels width) to reduce the data used in the least squares solution and improve processing time. Only data within the defined region of interest are used to obtain the least squares solution. Once the parameters for the model are determined, the flow at individual points can be determined using Eq. (1). Using the model it is possible to calculate the change in position of any *x*–*y* point from one image to the next using the affine transformation. This includes points outside of the region of interest and thus, the fascicle endpoints can be outside the region of interest. In our algorithm, the calculated affine transformation is applied to the Cartesian coordinates of the defined fascicle end points from the first image frame of each pair to calculate the new coordinates in the subsequent image. This iterative approach allows