Direct Linear Transformation for the Measurement of *In-Situ* Peripheral Nerve Strain During Stretching

Virginia Orozco[1], **Sriram Balasubramanian**[1], **Anita Singh**[2]

[1] School of Biomedical Engineering, Science and Health Systems, Drexel University [2] Bioengineering Department, Temple University

Corresponding Author

Anita Singh
anita.singh0001@temple.edu

Citation

Orozco, V., Balasubramanian, S., Singh, A. Direct Linear Transformation for the Measurement of *In-Situ* Peripheral Nerve Strain During Stretching. *J. Vis. Exp.* (203), e65924, doi:10.3791/65924 (2024).

Date Published

January 12, 2024

DOI

10.3791/65924

URL

jove.com/video/65924

Abstract

Peripheral nerves undergo physiological and non-physiological stretch during development, normal joint movement, injury, and more recently while undergoing surgical repair. Understanding the biomechanical response of peripheral nerves to stretch is critical to the understanding of their response to different loading conditions and thus, to optimizing treatment strategies and surgical interventions. This protocol describes in detail the calibration process of the stereo-imaging camera system via direct linear transformation and the tracking of the three-dimensional *in-situ* tissue displacement of peripheral nerves during stretch, obtained from three-dimensional coordinates of the video files captured by the calibrated stereo-imaging camera system.

From the obtained three-dimensional coordinates, the nerve length, change in the nerve length, and percent strain with respect to time can be calculated for a stretched peripheral nerve. Using a stereo-imaging camera system provides a non-invasive method for capturing three-dimensional displacements of peripheral nerves when stretched. Direct linear transformation enables three-dimensional reconstructions of peripheral nerve length during stretch to measure strain. Currently, no methodology exists to study the *in-situ* strain of stretched peripheral nerves using a stereo-imaging camera system calibrated via direct linear transformation. Capturing the *in-situ* strain of peripheral nerves when stretched can not only aid clinicians in understanding underlying injury mechanisms of nerve damage when overstretched but also help optimize treatment strategies that rely on stretch-induced interventions. The methodology described in the paper has the potential to enhance our understanding of peripheral nerve biomechanics in response to stretch to improve patient outcomes in the field of nerve injury management and rehabilitation.

Introduction

Peripheral nerves (PNs) undergo stretch during development, growth, normal joint movement, injury, and surgery[1]. PNs display viscoelastic properties to protect the nerve during regular movements[2,3] and maintain the structural health of its nerve fibers[2]. Because PN response to mechanical stretch has been shown to depend on the type of nerve fiber damage[4], injuries to adjacent connective tissues[2,4], and testing approaches (i.e., loading rate or direction)[5,6,7,8,9,10,11,12,13,14], it is essential to distinguish the biomechanical responses of PNs during normal range of motion versus non-physiological range at both slow- and rapid-stretch rates. This can further the understanding of the PN injury mechanism in response to stretch and aid in timely and optimized intervention[1,4,15,16]. There has been a growing trend in physical therapy to evaluate and intervene based on the relationship between nerve physiology and biomechanics[17]. By understanding the differences in PN biomechanics at various applied loads, physical therapists can be better prepared to modify current interventions[17].

Available biomechanical data of PNs in response to stretch remains variable and can be attributed to testing equipment and procedures and differences in elongation data analysis[5,6,7,8,9,10,11,12,13,14,16]. Furthermore, measuring three-dimensional (3D) *in-situ* nerve displacement remains poorly described in the currently available literature. Previous studies have used stereo-imaging techniques to maximize the accuracy of 3D reconstruction of tissue displacement of facet joint capsules[18,19]. The direct linear transformation (DLT) technique enables the conversion of two or more two-dimensional (2D) views to 3D real-world coordinates (i.e., in mm)[20,21,22]. DLT provides a high-accuracy calibration method for stereo-imaging camera systems because it enables precise reconstruction of 3D positions, accounting for lens distortion, camera parameters, and image coordinates, and permits flexibility in stereo-imaging camera setup[20,21,22]. Studies using DLT-calibrated stereo-imaging camera systems are typically used to study locomotion and gait analysis[22,23]. This protocol aims to offer a detailed methodology to determine the *in-situ* strain of PNs at varying degrees of stretch using a DLT-calibrated stereo-imaging camera system and an open-source tracking software[22].

Protocol

All procedures described were approved by the Drexel University Institutional Animal Care and Use Committee (IACUC). The neonatal piglet was acquired from a United States Department of Agriculture (USDA)-approved farm located in Pennsylvania, USA.

1. Stereo-imaging system setup

1. Attach a stereo-imaging camera system that captures up to 100 frames/s (FPS) to a utility stand. The stereo-imaging camera system used in this study is a passive stereo camera with two horizontally aligned cameras (referred to as the left and right cameras) separated by a baseline of 63 mm (**Figure 1**).

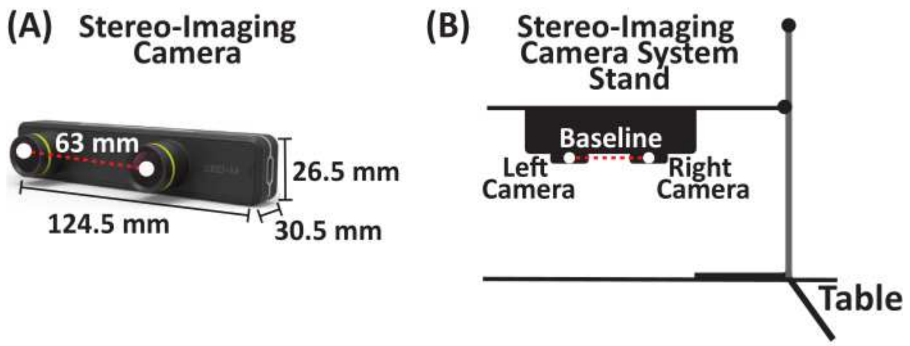
(A) Stereo-Imaging Camera
63 mm
26.5 mm
124.5 mm
30.5 mm
(B) Stereo-Imaging Camera System Stand
Baseline
Left Camera
Right Camera
Table

Figure 1: Stereo-imaging camera system. (**A**) Parallel stereo-imaging camera system with two cameras (left and right cameras) separated by a baseline of 63 mm. (**B**) Schematic of stereo-imaging camera system and stand setup. Please click here to view a larger version of this figure.

2. Stereo-imaging system DLT Calibration-digitizing the 3D control volume

1. Obtain three clear acrylic plexiglass square sheets (12 in x 12 in x 0.125 in). On each sheet, place a grid and draw at least 10 points, resulting in a minimum of 30 points distributed across the 3D control volume in the x, y, and z coordinate planes. Construct the 3D control volume by stacking the three sheets at varying heights to capture the maximum height of what will be recorded (**Figure 2A**).

2. Digitize all the points on the 3D control volume using a digitizer with a foot pedal. Acquire x, y, and z coordinates (in mm) (**Figure 2B**) by establishing the origin (0, 0, 0) on the 3D control cube, defining the positive x- and y-directions, opening a document to save the digitized (x, y, z) coordinates (in mm) of each point, and saving (x, y, z) coordinates (in mm) as a *.csv file.

   NOTE: The (x, y, z) coordinates are relative to the set origin on the 3D control cube.

3. Use these digitized (x, y, z) coordinates (in mm) to calibrate the stereo-imaging camera system's left and right cameras, respectively.

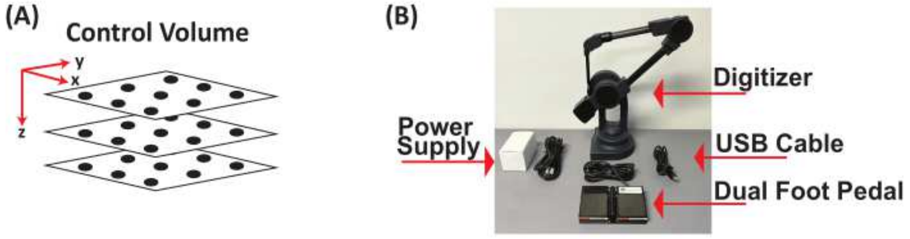

Figure 2: Three-dimensional control volume and digitizer with foot pedal. (**A**) Schematic of 3D control volume. (**B**) Components of digitizer with foot pedal used to digitize 3D control volume to obtain (x, y, z) coordinates in mm. Abbreviation: 3D = three-dimensional. Please click here to view a larger version of this figure.

3. Stereo-imaging camera system calibration-generation of direct linear transformation coefficients

1. Attach the stereo-imaging camera system to a utility stand (**Figure 3A**).
2. Position the stereo-imaging camera system 6 cm above the 3D control volume (**Figure 3A**).
3. Connect the stereo-imaging camera system to a laptop via a USB type-C cable.
4. Open the imaging software (see the **Table of Materials**).
5. Image the 3D control volume. The output image (**Supplemental Figure S1**) includes both the left and right camera views (**Figure 3B**).
6. Run the custom MATLAB code (**Supplemental File 1**) to separate the output image into two images, left and right images, respectively (**Figure 3C**, **Supplemental Figures S2**, and **Supplemental Figures S3**, respectively).
7. Click **Run** to initialize the **DLTcal5.m GUI**[22] (**Supplemental File 2**).
    1. Click **initialize** on the **DLTcal5 controls** window to select the *.csv file with the digitized (x, y, z) coordinates (in mm) (**Figure 4A** and **Supplemental File 3**).
    2. Select the corresponding image of the 3D control volume from the first view of the stereo-imaging camera system (**Supplemental Figure S2**). For this stereo-imaging camera system, the left camera view corresponds to the first camera view (**Figure 4B**).
    3. The first camera view image (i.e., left camera view) pops up.
    4. Select the points in the order the points were digitized in Section 2 to obtain the 2D pixel coordinates from the left camera view (**Figure 4B**).
    5. Set **current point** on the **DLTcal5 Controls** window and click on the corresponding **current point** on the loaded first camera view (i.e., left camera view) image.
    6. After selecting all the points on the loaded first camera view, click **compute coefficients** to generate the 11 DLT coefficients for the left camera view (**Figure 4B**).
    7. Click **add a camera** on the **DLTcal5 controls** window and repeat steps 3.7.2-3.7.6 to generate the 11 DLT coefficients for the right camera view (i.e., second camera view) (**Figure 4B, C** and **Supplemental Figure S3**).
    8. Click **save data** on the **DLTcal5 controls** window to select the folder where the output files will be saved (**Figure 4D**).
    9. The output files include the 2D (x, y) pixel coordinates (**Supplemental File 4**) and corresponding 11 DLT coefficients for the left and right camera views of the stereo-imaging camera system (**Figure 4D** and **Supplemental File 5**).
    10. The stereo-imaging camera system is calibrated.

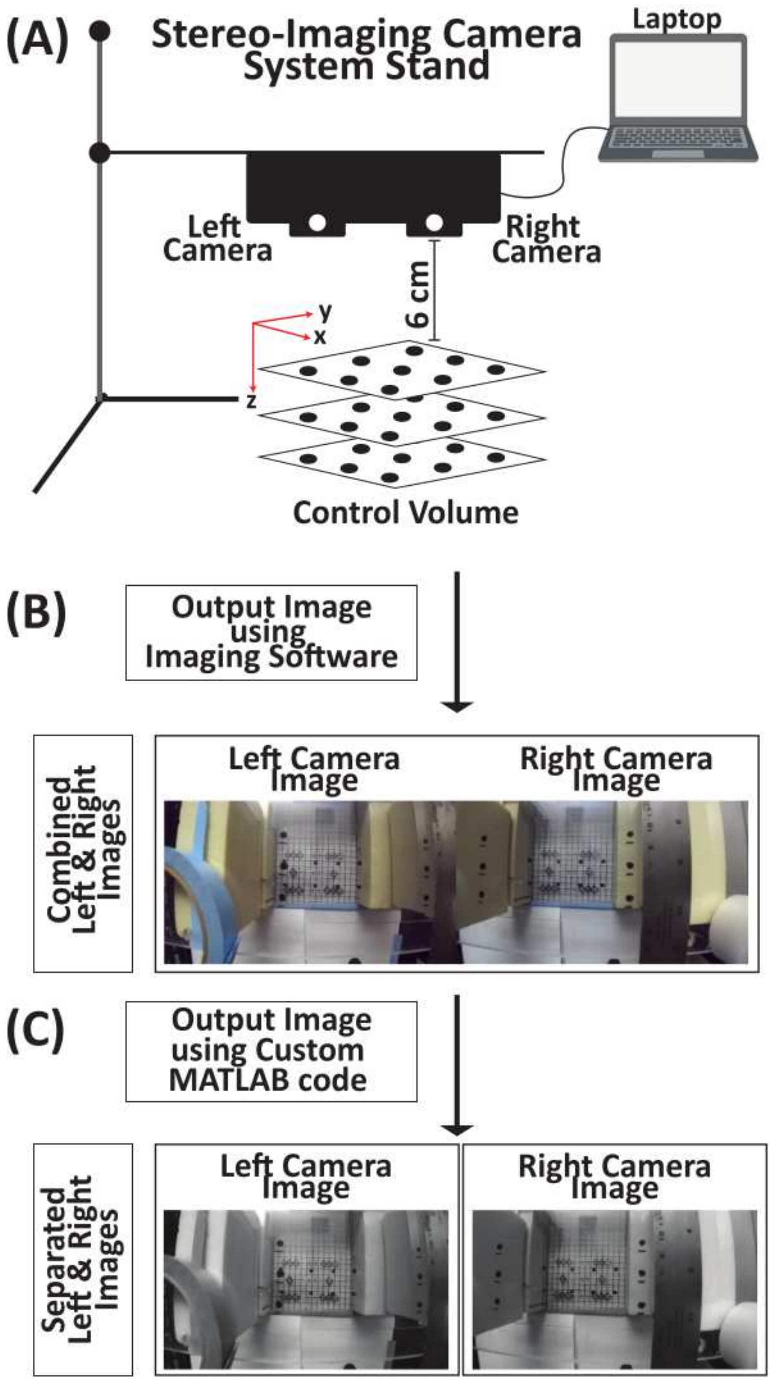
(A)
Stereo-Imaging Camera System Stand
Laptop
Left Camera
Right Camera
6 cm
y
x
z
Control Volume
(B)
Output Image using Imaging Software
Combined Left & Right Images
Left Camera Image
Right Camera Image
(C)
Output Image using Custom MATLAB code
Separated Left & Right Images
Left Camera Image
Right Camera Image

Figure 3: Schematic for acquiring an image of three-dimensional control volume using a stereo-imaging camera system for direct linear transformation calibration. (**A**) Attach the stereo-imaging camera system to a stand and then connect it to a laptop via a USB type-C cable. Place the 3D control volume 6 cm under the stereo-imaging camera system. (**B**) Using the imaging software, take an image of the 3D control volume. The output image is a combined image from the left and right cameras. (**C**) Using a custom MATLAB code, the combined output image is separated into individual left and

right images of the 3D control volume. Abbreviation: 3D = three-dimensional. Please click here to view a larger version of this figure.

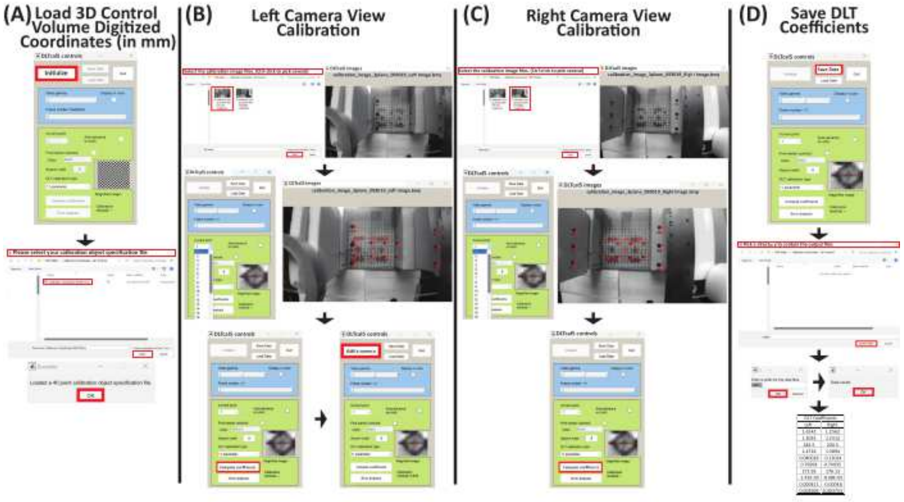
(A) Load 3D Control Volume Digitized Coordinates (in mm)
(B) Left Camera View Calibration
(C) Right Camera View Calibration
(D) Save DLT Coefficients

Figure 4: Schematic for generating direct linear transformation coefficients for left and right camera views of a stereo-camera imaging system. (**A**) Run **DLTcal5.m**[22], click **initialize** on the **controls** window, and select the ***.csv file** with the digitized (x, y, z) coordinates (in mm) of the 3D control volume. (**B**) Select the calibration image of the left camera view. Then, select the points on the image in the same order that they were digitized. Then, click **compute coefficients** to generate the DLT coefficients for the left camera view. Next, click **Add camera** to repeat the steps for the right camera view. (**C**) Select the calibration image of the right camera view. Then, select the points on the image in the same order that they were digitized. Then, click **compute coefficients** to generate the DLT coefficients for the right camera view. (**D**) Click **Save Data** to select the directory to save the DLT coefficients for the left and right camera views. Enter the name for the output file and click **OK** and the DLT coefficients are saved as a *.csv file. Abbreviation: 3D = three-dimensional and DLT = direct linear transformation. Please click here to view a larger version of this figure.

4. Data acquisition

1. Place anesthetized neonatal Yorkshire piglet (3-5 days old) in a supine position with upper limbs in abduction to expose the axillary region. Make a midline incision through the skin and fascia overlying the trachea down to the upper third of the sternum.
2. Use blunt dissection to expose the brachial plexus nerves.
3. Squirt saline solution on the exposed brachial plexus nerve to keep them hydrated before, during, and after testing.
4. Cut the distal end of a brachial plexus PN and clamp to the mechanical testing apparatus.

5. Attach the stereo-imaging camera system to the utility stand, place it up to 6 cm above the PN to be stretched, and then connect the stereo-imaging camera system to a laptop via a USB type-C cable (**Figure 5A**).

6. Use an ink-based skin marker to place markers on the insertion and clamp sites and an additional two to four markers along the length of the PN depending on the nerve length, for displacement tracking (**Figure 5B**).

7. Place a calibration grid (i.e., laminated grid with 0.5 cm x 0.5 cm squares) and a 1 cm ruler, flat underneath the PN for data analysis (**Figure 5B**).

8. Record the initial length of the PN after clamping and just before stretching.

9. Stretch the PN at a displacement rate of 500 mm/min to failure or a predetermined stretch.

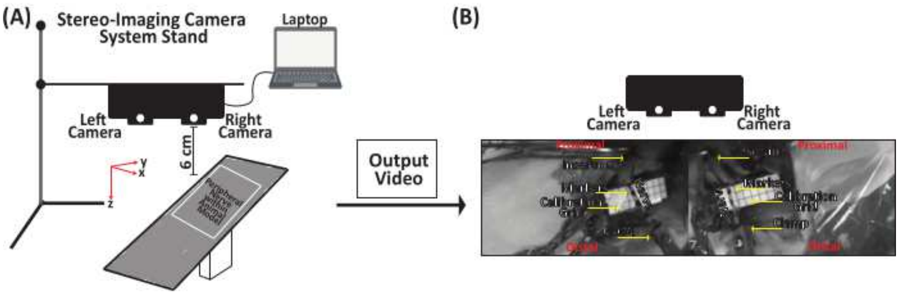

Figure 5: Representative schematic for data acquisition of peripheral nerve stretching. (**A**) Attach the stereo-imaging camera system to a stand and then connect it to a laptop via a USB type-C cable. Place the stereo-imaging camera system up to 6 cm above the peripheral nerve. (**B**) The peripheral nerve is clamped to the mechanical setup at the distal end. Using an ink-based skin marker, place a marker on the insertion and clamp sites and an additional two to four markers along the nerve length. Saline is squirted on the peripheral nerve to keep it hydrated before, during, and after testing. Please click here to view a larger version of this figure.

5. Data analysis-marker trajectory tracking

1. Run the custom MATLAB code (**Supplemental File 6**) to separate the output video file (**Supplemental File 7**) into two video files, left and right camera video files (**Supplemental File 8** and **Supplemental File 9**, respectively).

2. Click **Run** to initialize the **DLTdv7.m**[22] GUI (**Supplemental File 10**).

   1. The DLTdv7 controls window pops up and the **new project, load project**, and **quit** buttons are enabled (**Figure 6A**).

   2. Click **new project** on the **DLTdv7 controls** window to begin a new project.

   3. When a dialog box pops up, select **2** to indicate two video files (i.e., left and right camera views) to track the displacement marker trajectories on a stretched peripheral nerve (**Figure 6B**).

4. Select the first video file (i.e., **Video 1**), which is the video file from the left camera view (**Supplemental File 8**), and click **open** (**Figure 6C**). Then, select the second video file (i.e., **Video 2**), which is the video file from the right camera view (**Supplemental File 9**), and click **open** (**Figure 6C**).

5. After the two video files are selected, click **yes** to signify the video files were acquired from camera views calibrated via DLT.

6. Select the corresponding DLT coefficients *.csv file (**Supplemental File 5**) for the stereo-imaging camera system and click on **open** (**Figure 6D**).

7. The initial video frames are displayed from both video files and the rest of the **DLTdv7 controls** window is activated. The **new project** button is replaced with the **recompute 3D points** button and the **load project** button is replaced with the **save** button (**Figure 6E**).

8. On the **DLTdv7 controls** window, ensure the **frame number** is on **1**, the **current point** is set to **1**, **autotrack mode** is **off**, and **update all videos**, **DLT visual feedback**, and **show 2D tracks** are checked (**Figure 6E**).

9. Ensure that the tracking points are placed on the displacement markers of the PN such that the **insertion marker** corresponds to point **1**, **marker 1** corresponds to point **2**, and so on with the clamp marker being the final point.

10. Place point **1** on the **insertion marker** on **Video 1** (i.e., left camera view video file) ensuring the placed point is at the center of the insertion marker. Use keyboard shortcuts (**Table 1**) to move the placed point to the center of the insertion marker (**Figure 7A**).

11. Because **DLT visual feedback** is checked, when a point is placed in **Video 1**, a blue epipolar line appears in **Video 2** (i.e., right camera view video file) (**Figure 7**). Place point **1** on the insertion marker in **Video 2** using the blue epipolar line as reference. Use the keyboard shortcuts (**Table 1**), as needed, to move the placed point to the center of the insertion marker (**Figure 7A**).

12. Click **add a point** on the **DLTdv7 controls** window to add points on the other tissue markers to track their trajectories. Refer to the **current point** on the **DLTdv7 controls** window to know which point is active.

13. Click **add a point**. Place point **2** on **marker 1** in **Video 1**. Use the blue epipolar line and keyboard shortcuts to place point **2** on **marker 1** in **Video 2**. Continue adding and placing points, first on **Video 1** and then on **Video 2**, for all displacement markers along the length of the nerve between the insertion and clamp (i.e., final point) (**Figure 7B-F**).

14. Once all initial points are placed in **Video 1** and **Video 2** (i.e., the left and right camera video files, respectively), ensure the **frame number** is on **1** and the **current point** is set to **1** on the **DLTdv7 controls** window.

15. On the **DLTdv7 controls** window, change **autotrack mode** to **auto-advance** from the dropdown menu and **autotrack predictor** to **extended Kalman** from the dropdown menu (**Figure 8A**).

16. Complete the tracking of all the placed points in **Video 1** first and then complete tracking in **Video 2**. Track the marker trajectory by left-clicking on the center of the marker frame by frame until failure (i.e., frame before gross rupture of the peripheral nerve) or the entirety of the video for predetermined stretch is achieved.

17. Begin tracking point **1** in **Video 1**. Zoom in and out (**Table 1**) as desired to ensure tracking is at the center of the marker; click frame by frame until the failure or end of the video (**Figure 8B**). After completing the tracking of point **1** in **Video 1**, return to **frame 1** and change the **current point** to point **2** from the dropdown menu on the **DLTdv7 controls** window. The previous tracked point will turn light blue and its trajectory will turn yellow. The current point will have a green diamond and a pink center.

18. Complete the tracking of all the points in **Video 1** by left-clicking frame by frame for each point until failure or end of the video (**Figure 8C-G**).

19. In **Video 2**, use the blue epipolar line to track points in reference to **Video 1** (**Figure 9**). On the **DLTdv7 controls** window, return **frame number** to **1** and set **current point** to **1**, and begin tracking point 1's trajectory in **Video 2**.

20. Follow the same steps (5.2.16-5.2.18) to track the remaining points in **Video 2**.

21. After tracking is complete in **Video 1** and **Video 2**, click **export points** on the **DLTdv7 controls** window to export the (x, y, z) coordinates (in mm) of the tracked points.

22. A dialog box pops up to select the directory where to save the output files. Click on the **directory location**.

23. Another dialog box pops up to set the name of the output files. Set the output files' name (i.e., nerve1_101-Jan-2001videoanalyzed_cal09.30_trial1_).

24. Another dialogue box pops up. Select the save **format** as **flat**.

25. Another dialog box pops up. Select no to calculate the 95% confidence interval.

26. A final dialog box pops up that shows the data is exported and saved, and the four output files are exported to the selected directory location (**Supplemental File 11, Supplemental File 12, Supplemental File 13,** and **Supplemental File 14**).

27. Click **save project** on the **DLTdv7 controls** window to save the current project (**Supplemental File 15**) in the same directory as the output files.

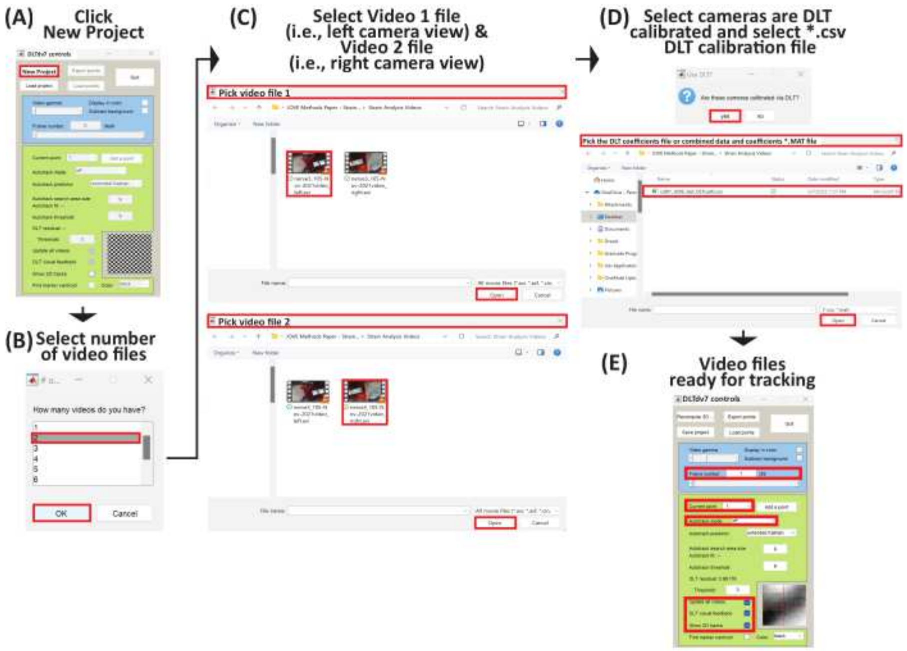
(A) Click New Project
DLTdv7 controls
New Project
(B) Select number of video files
How many videos do you have?
OK
Cancel
(C) Select Video 1 file (i.e., left camera view) & Video 2 file (i.e., right camera view)
Pick video file 1
Pick video file 2
Open
Cancel
(D) Select cameras are DLT calibrated and select *.csv DLT calibration file
Use DLT?
yes
No
Pick the DLT coefficients file or combined data and coefficients *.MAT file
(E) Video files ready for tracking
DLTdv7 controls

Figure 6: Schematic to set up a new project to begin three-dimensional trajectory tracking. (**A**) Run **DLTdv7.m**[22] and click **New Project** to begin a new project. (**B**) Select **2** as the number of video files. (**C**) Select **Video 1 file** (i.e., left camera view) and then select **Video 2** file (i.e., right camera view). (**D**) Select **yes** as the video files come from a DLT calibrated stereo-imaging camera system. Then, select the *.csv file containing the DLT coefficients. (**E**) The selected video files are now ready for tracking. Please click here to view a larger version of this figure.

Key/Click	Description
Left Click	Tracks trajecgtory of a point in frame clicked
(+) Key	Zooms current video frame **in** arount mosue pointer
(-) Key	Zooms current video frame **out** arount mosue pointer
(i) Key	Move point **up**
(j) Key	Move point **left**
(k) Key	Move point **right**
(m) Key	Move point **down**

Table 1: Keyboard and mouse shortcuts for tracking point trajectory.

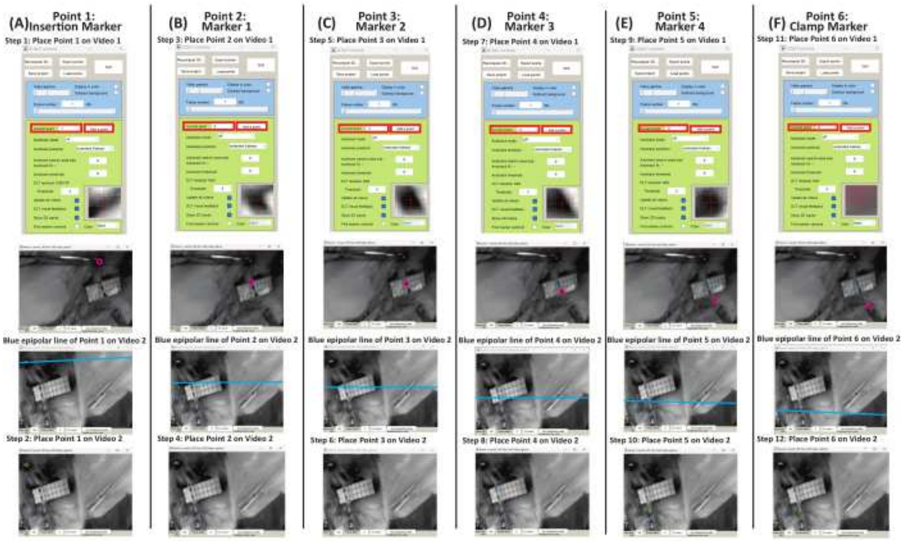
(A) Point 1: Insertion Marker
Step 1: Place Point 1 on Video 1
Blue epipolar line of Point 1 on Video 2
Step 2: Place Point 1 on Video 2
(B) Point 2: Marker 1
Step 3: Place Point 2 on Video 1
Blue epipolar line of Point 2 on Video 2
Step 4: Place Point 2 on Video 2
(C) Point 3: Marker 2
Step 5: Place Point 3 on Video 1
Blue epipolar line of Point 3 on Video 2
Step 6: Place Point 3 on Video 2
(D) Point 4: Marker 3
Step 7: Place Point 4 on Video 1
Blue epipolar line of Point 4 on Video 2
Step 8: Place Point 4 on Video 2
(E) Point 5: Marker 4
Step 9: Place Point 5 on Video 1
Blue epipolar line of Point 5 on Video 2
Step 10: Place Point 5 on Video 2
(F) Point 6: Clamp Marker
Step 11: Place Point 6 on Video 1
Blue epipolar line of Point 6 on Video 2
Step 12: Place Point 6 on Video 2

Figure 7: Schematic to place initial points on tissue markers for Video 1 and Video 2 using DLTdv7.m[22]. (**A**) Set **current point** to **1**. Place point **1** on the **insertion marker** in **Video 1**. Using the blue epipolar line in **Video 2**, place point **1** on the **insertion marker**. (**B**) Set **current point** to **2**. Place point **2** on **marker 1** in **Video 1**. Using the blue epipolar line in **Video 2**, place point **2** on **marker 1**. (**C**) Set **current point** to **3**. Place point **3** on **marker 2** in **Video 1**. Using the blue epipolar line in **Video 2**, place point **3** on **marker 2**. (**D**) Set **current point** to **4**. Place point **4** on **marker 3** in **Video 1**. Using the blue epipolar line in **Video 2**, place point **4** on **marker 3**. (**E**) Set **current point** to **5**. Place point **5** on **marker 4** in **Video 1**. Using the blue epipolar line in **Video 2**, place point **5** on **marker 4**. (**F**) Set **current point** to **6**. Place point **6** on the **clamp marker** in **Video 1**. Using the blue epipolar line in **Video 2**, place point **6** on the **clamp marker**. Please click here to view a larger version of this figure.

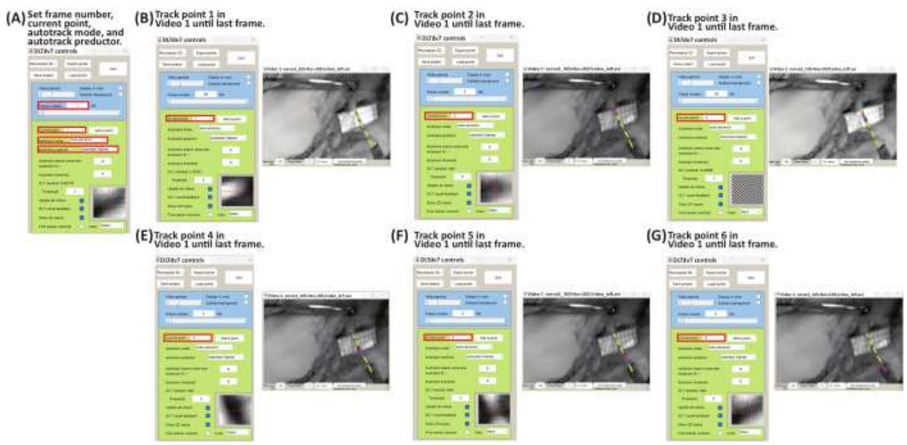
(A) Set frame number, current point, autotrack mode, and autotrack preductor.
DLTdv7 controls
(B) Track point 1 in Video 1 until last frame.
DLTdv7 controls
(C) Track point 2 in Video 1 until last frame.
DLTdv7 controls
(D) Track point 3 in Video 1 until last frame.
DLTdv7 controls
(E) Track point 4 in Video 1 until last frame.
DLTdv7 controls
(F) Track point 5 in Video 1 until last frame.
DLTdv7 controls
(G) Track point 6 in Video 1 until last frame.
DLTdv7 controls

Figure 8: Schematic for tracking marker point trajectories of Video 1 using DLTdv7.m[22]. (**A**) Set **frame number** to **1**, **current point** to **1**, **autotrack mode** to **auto-advance**, and **autotrack predictor** to **extended Kalman**. (**B**) Set **current point** to **1**. On **Video 1** file, begin tracking the insertion marker (i.e., point **1**) displacement by left-clicking frame-by-frame until the last frame. (**C**) Set **frame number** to **1** and **current point** to **2**. On **Video 1** file, begin tracking marker 1 (i.e., point **2**) displacement by left-clicking frame-by-frame until the last frame. (**D**) Set **frame number** to **1** and **current point** to **3.** On **Video 1** file, begin tracking marker 2 (i.e., point **3**) displacement by left-clicking frame-by-frame until the last frame. (**E**) Set **frame number** to **1** and **current point** to **4**. On **Video 1** file, begin tracking marker 3 (i.e., point **4**) displacement by left-clicking frame-by-frame until the last frame. (**F**) Set **frame number** to **1** and **current point** to **5**. On **Video 1** file, begin tracking marker 4 (i.e., point **5**) displacement by left-clicking frame-by-frame until the last frame. (**G**) Set **frame number** to **1** and **current point** to **6**. On **Video 1** file, begin tracking the clamp marker (i.e., point **6**) displacement by left-clicking frame-by-frame until the last frame. Please click here to view a larger version of this figure.

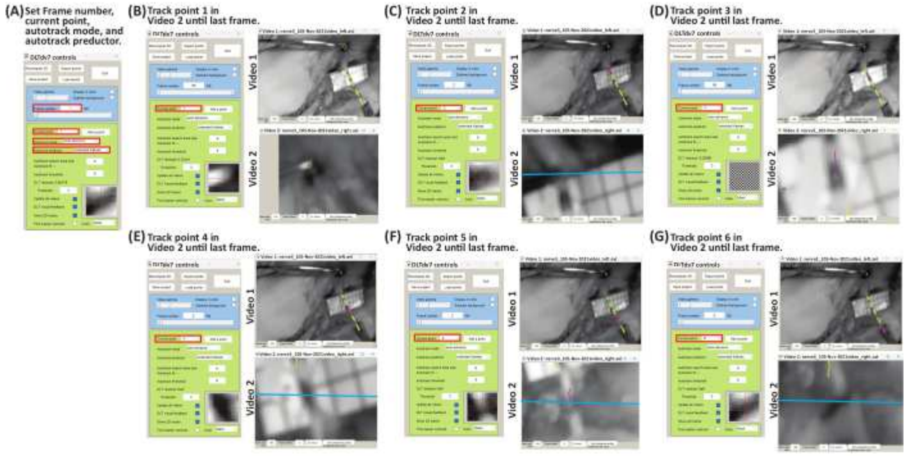
(A) Set Frame number, current point, autotrack mode, and autotrack predictor.
DLTdv7 controls
(B) Track point 1 in Video 2 until last frame.
(C) Track point 2 in Video 2 until last frame.
(D) Track point 3 in Video 2 until last frame.
(E) Track point 4 in Video 2 until last frame.
(F) Track point 5 in Video 2 until last frame.
(G) Track point 6 in Video 2 until last frame.
Video 1
Video 2

Figure 9: Schematic for tracking marker point trajectories of Video 2 using DLTdv7.m[22]. (**A**) Set **frame number** to **1**, **current point** to **1**, **autotrack mode** to **auto-advance**, and **autotrack predictor** to **extended Kalman**. (**B**) Set **current point** to **1**. Using the blue epipolar line on **Video 2** file, begin tracking the insertion marker (i.e., point **1**) displacement by left-clicking frame-by-frame until the last frame. (**C**) Set **frame number** to **1** and **current point** to **2**. Using the blue epipolar line on **Video 2** file, begin tracking marker 1 (i.e., point **2**) displacement by left-clicking frame-by-frame until the last frame. (**D**) Set **frame number** to **1** and **current point** to **3**. Using the blue epipolar line on **Video 2** file, begin tracking marker 2 (i.e., point **3**) displacement by left-clicking frame-by-frame until the last frame. (**E**) Set **frame number** to **1** and **current point** to **4**. Using the blue epipolar line on **Video 2** file, begin tracking marker 3 (i.e., point **4**) displacement by left-clicking frame-by-frame until the last frame. (**F**) Set **frame number** to **1** and **current point** to **5**. Using the blue epipolar line on **Video 2** file, begin tracking marker 4 (i.e., point **5**) displacement by left-clicking frame-by-frame until the last frame. (**G**) Set **frame number** to **1** and **current point** to **6**. Using the blue epipolar line on **Video 2** file, begin tracking the clamp marker (i.e., point **6**) displacement by left-clicking frame-by-frame until the last frame. Please click here to view a larger version of this figure.

6. Data analysis-strain analysis

1. Run a custom MATLAB code (**Supplemental File 16**) to import the tracked 3D (x, y, z) marker trajectories (in mm).

2. On the MATLAB command window, type: percentStrain, deltaLi, lengthNi, filename] = PercentStrain_3D

3. Enter the **rupture time**, for example, if the video files have 59 frames, the time is **0.59 s**; enter the **number of tracked points**, and select the ***_xyzpts.csv file**, with the tracked 3D (x, y, z) trajectories (in mm).

4. Select the **directory** to save the output length vs time (**Supplemental Figure S4**), change in length vs time (**Supplemental Figure S5**), and strain vs time (**Supplemental Figure S6**) plots and ***.xls file** with time,

length, change in length, and strain (**Supplemental File 17**).

5. Calculate the length (l), change in length (Δl), and percent strain using equations **1-3**:

$$l_i = \sqrt{\left(x_{2_i} - x_{1_i}\right)^2 + \left(y_{2_i} - y_{1_i}\right)^2 + \left(z_{2_i} - z_{1_i}\right)^2} \qquad (\mathbf{1})$$

Where l_i is the distance between any two markers at any time point; x_{1i}, y_{1i}, z_{1i} are the 3D coordinates of one of the two markers; and x_{2i}, y_{2i}, z_{2i} are the 3D coordinates of the second markers.

$$\Delta l_i = l_i - l_o \qquad (\mathbf{2})$$

Where l_i is the distance between any two markers at any time point, and l_o is the distance between any two markers at the original/zero-time point.

$$percent\ strain_i = \frac{\Delta l_i}{l_o} \times 100 \qquad (\mathbf{3})$$

Where Δl_i is the change in length between two markers at any time point, and l_o is the distance between any two markers at the original/zero-time point.

Representative Results

Using the described methodology, various output files are obtained. The DLTdv7.m *_xyzpts.csv (**Supplemental File 12**) contains the (x, y, z) coordinates in millimeters of each tracked point at each time frame that is further used to calculate the length, change in length, and strain of the stretched PN. Representative length-time, change in length-time, and strain-time plots of a stretched PN are shown in **Figure 10**. The stretched PN had an insertion marker, four markers along its length, and a clamp marker comprising six segments. By quantifying the overall and segmental strains of the stretched PN, a better understanding is gained of the non-homogeneity of these structures as well as the segmental contribution to the overall stretch. The length-time (**Figure 10A**) and change in length-time plots (**Figure 10B**) are used to calculate the strain-time plots (**Figure 10C**). In addition to the plots, a spreadsheet (**Supplemental File 17**) with the plot data (i.e., time, length, change in length, and strain) is exported.

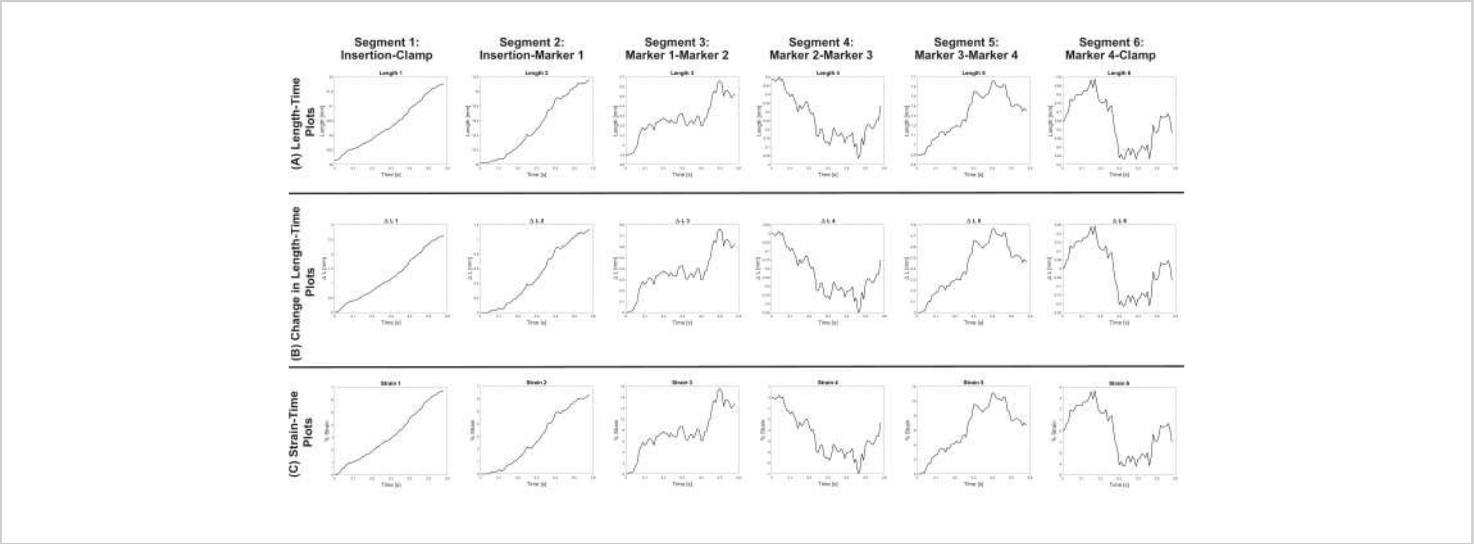

Figure 10: Representative plots of strain-time, change in length-time, and length-time of a stretched peripheral nerve. (**A**) Length-time plots of the entire nerve (i.e., segment 1) and for all segments between adjacent markers (i.e.,

segments 2-6). (**B**) Change in length-time plots of the entire nerve (i.e., segment 1) and for all segments between adjacent markers (i.e., segments 2-6). (**C**) Strain-time plots of the entire nerve (i.e., segment 1) and for all segments between adjacent markers (i.e., segments 2-6). Please click here to view a larger version of this figure.

The other three DLTdv7.m output files (**Supplemental File 11, Supplemental File 13**, and **Supplemental File 14**) and output project (**Supplemental File 15**) are used to reload the project in case the marker points' trajectories need to be retracked.

Supplemental Figure S1: Output image of three-dimensional (3D) control volume. Output image of 3D control volume taken using the parallel stereo-imaging camera system and imaging software system for direct linear transformation calibration. Please click here to download this File.

Supplemental Figure S2: Left image of three-dimensional (3D) control volume. Left image of 3D control volume used for direct linear transformation calibration. Please click here to download this File.

Supplemental Figure S3: Right image of three-dimensional (3D) control volume. Right image of 3D control volume used for direct linear transformation calibration. Please click here to download this File.

Supplemental Figure S4: Length-time output plots. (**A-F**) Length-time output plots of each segment at each time frame. Please click here to download this File.

Supplemental Figure S5: Change in length-time output plots. (**A-F**) Change in length-time output plots of each segment at each time frame. Please click here to download this File.

Supplemental Figure S6: Strain-time output plots. (**A-F**) Length-time output plots of each segment at each time frame. Please click here to download this File.

Supplemental File 1: Custom MATLAB code crop_left_right_stereoimage.m. Custom MATLAB code used to separate the output image into two images, left and right images, respectively. Please click here to download this File.

Supplemental File 2: DLTcal5.m[22]. Open-source MATLAB code used to obtain the direct linear transformation coefficients of the parallel stereo-imaging camera system. Please click here to download this File.

Supplemental File 3: Three-dimensional (3D) Control Volume Digitized Points. Spreadsheet file (3D Control Volume_Digitized Pts.csv) containing the digitized (x, y, z) points in millimeters of the points on the 3D control volume. Please click here to download this File.

Supplemental File 4: DLTcal5.m output spreadsheet file containing the (x, y) pixel coordinates of the three-dimensional (3D) control volume points. Output spreadsheet file (cal01_JOVE_test_xypts.csv) containing the (x, y) pixel coordinates of the 3D control volume points using DLTcal5.m[22]. Please click here to download this File.

Supplemental File 5: DLTcal5.m output spreadsheet file containing the 11 direct linear transformation (DLT) coefficients. Output spreadsheet file (cal01_JOVE_test_DLTcoefs.csv) containing the 11 DLT coefficients for the left and right camera views of the stereo-imaging camera system using DLTcal5.m[22]. Please click here to download this File.

Supplemental File 6: Custom MATLAB code crop_left_right_stereovideo.m. Custom MATLAB code used to separate the output video file into two video files, left and right camera video files. Please click here to download this File.

Supplemental File 7: Output video file of a stretched peripheral nerve. Output video file (nerve3_105-Nov-2021video.avi) of a stretched peripheral nerve containing the combined video files of the left and right camera views. Please click here to download this File.

Supplemental File 8: Left camera view video file (i.e., Video 1) of a stretched peripheral nerve. Left camera view video file (nerve3_105-Nov-2021video_left.avi) of a stretched peripheral nerve used to track marker point trajectories using DLTdv7.m[22]. Please click here to download this File.

Supplemental File 9: Right camera view video file (i.e., Video 2) of a stretched peripheral nerve. Right camera view video file (nerve3_105-Nov-2021video_right.avi) of a stretched peripheral nerve used to track maker point trajectories using DLTdv.7.m[22]. Please click here to download this File.

Supplemental File 10: DLTdv7.m[22]. Open source MATLAB code used to track marker point trajectories of video files obtained from a stereo-imaging camera system calibrated using direct linear transformation. Please click here to download this File.

Supplemental File 11: DLTdv7.m output file *_xypts.csv. The first DLTdv7.m output file is *_xypts.csv (nerve3_105-Nov-2021videoanalyzed_cal09.30_trial1_xypts.csv) contains the pixel coordinates (x1, y1), (x2, y2), etc…for each tracked point at each time frame. Please click here to download this File.

Supplemental File 12: DLTdv7.m output file *_xyzpts.csv. The second DLTdv7.m output file is *_xyzpts.csv (nerve3_105-Nov-2021videoanalyzed_cal09.30_trial1_xyzpts.csv) contains the real-world coordinates in millimeters (x1, y1, z1), (x2, y2, z2), etc…for each tracked point at each time frame. Please click here to download this File.

Supplemental File 13: DLTdv7.m output file *_xyzres.csv. The third DLTdv7.m output file is *_xyzres.csv (nerve3_105-Nov-2021videoanalyzed_cal09.30_trial1_xyzres.csv) contains the DLT residual for each tracked point at each time frame. Please click here to download this File.

Supplemental File 14: DLTdv7.m output file *_offset.csv. The first DLTdv7.m output file is *_offset.csv (nerve3_105-Nov-2021videoanalyzed_cal09.30_trial1_offsets.csv) contains video 1 and video 2 offset for each tracked point at each time frame. Please click here to download this File.

Supplemental File 15: DLTdv7.m project output file *_dvProject.mat. The DLTdv7.m project output file (nerve3_105-Nov-2021videoanalyzed_cal09.30_trial1_dvProject.mat) contains the paths of the video files, all interface settings, all clicked marker point trajectories, and calibration information allowing for easy reload of project to make changes, if necessary. Please click here to download this File.

Supplemental File 16: Custom MATLAB code PercentStrain_3D.m. Custom MATLAB code used to calculate length, change in length, and percent strain of a stretched nerve between adjacent markers at each time point. Please click here to download this File.

Supplement File 17: PrecentStrain_3D.m output file *_3Dstrain.xls. Output file *_3Dstrain.xls (nerve3_105-Nov-2021_3Dstrain.xls) that contains time, length, change in length, and strain of each tracked point at each time frame. Please click here to download this File.

Discussion

Studies reporting biomechanical properties of peripheral nerves (PNs) because of stretch injury vary, and that variation can be attributed to testing methodologies such as testing equipment and elongation analysis[5,6,7,8,9,10,11,12,13,14,16,24]. This protocol offers a detailed description to capture three-dimensional (3D) tissue displacements of PNs during stretch using a direct linear transformation (DLT) calibrated stereo-imaging camera system. The following critical steps must be followed for robust strain analysis methodology. First, the 3D control volume must have a minimum of 30 points. Second, the 3D control volume points must be digitized in the same direction using the digitizer with the foot pedal and DLTcal5.m[22] program. Third, when taking an image of the 3D control volume, the stereo-imaging camera system must be placed at the maximum height of the object of interest being recorded (i.e., 6 cm for the current study). Fourth, users must know which are the left and right camera views. Finally, marker trajectory tracking must be completed on **Video 1** (i.e., left camera view) and then on **Video 2** (i.e., right camera view).

MATLAB 2019A or a newer version must be downloaded and the add-ons Image Acquisition Toolbox, Computer Vision Toolbox, Image Processing Toolbox, Curve Fitting Toolbox, Statistics and Machine Learning Toolbox, and Financial Analysis Toolbox must also be installed.

DLT is a versatile method that can be applied for camera calibration, 3D reconstruction, object tracking, and motion analysis[20,22,23,25,26]. Additionally, DLT provides a robust and accurate methodology for camera calibration because it allows for precise estimation of camera parameters, including intrinsic parameters (i.e., focal length) and extrinsic parameters (i.e., rotation, translation)[20,22]. Previous studies have used DLT-calibrated stereo-imaging camera systems to study locomotion and gait analysis of humans[23] or animal models[22]. The presented methodology can be easily implemented in any study investigating the biomechanical responses of PNs to stretch injury. This protocol provides open-source and custom MATLAB codes used and created, respectively, to calibrate a stereo-imaging camera system and track 3D tissue displacements of PNs during stretch.

In summary, a DLT-calibrated stereo-imaging camera system can provide a more comprehensive methodology to determine the strain of PNs in response to stretch. This current study aims to develop a methodology using a stereo-imaging camera and DLT to measure *in-situ* strain in stretched PNs. The development of such a method will aid in quantifying 3D strain in PNs during stretch. Studies investigating the failure of stretched PNs have shown that it results in maximum strain, persistent deformation, regional strain variation, and failure location, highlighting the importance of understanding the biomechanical response of PNs to stretching[9,27,28]. Additionally, studies have demonstrated that PNs undergo structural changes when subjected to stretch injuries[28,29]. Understanding the thresholds of PNs enduring these changes is essential, as it can provide insights into the resilience and integrity of stretched PNs[28,29]. Therefore, comprehensively quantifying the extent of stretch injury in PNs can help future studies correlate structural changes and functional deficits, providing a deeper understanding of the injury mechanism associated with PN stretching. Therefore, this current study is a critical first step to enhance the understanding of the injury biomechanics of PNs to help facilitate clinicians optimize treatment and rehabilitation to improve the patients' quality of life and alleviate suffering from PN injuries. Furthermore, the detailed protocol and supplemental files provided in the study can be incorporated into existing biomechanics teaching modules and enhance student learning of the DLT technique[30].

Disclosures

The authors have no conflicts of interest to disclose.

Acknowledgments

This research was supported by funding from the Eunice Kennedy Shriver National Institute of Child Health and Human Development of the National Institutes of Health under Award Number R15HD093024 and R01HD104910A and NSF CAREER Award Number 1752513.

References

1. Bueno, F. R., Shah, S. B. Implications of tensile loading for the tissue engineering of nerves. *Tissue Engineering Part B: Reviews.* **14** (3), 219-233 (2008).
2. Grewal, R., Xu, J., Sotereanos, D. G., Woo, S. L. Biomechanical properties of peripheral nerves. *Hand Clinics.* **12** (2), 195-204 (1996).
3. Papagiannis, G. et al. Biomechanical behavior and viscoelastic properties of peripheral nerves subjected to tensile stress: common injuries and current repair techniques. *Critical Reviews in Physical and Rehabilitation Medicine.* **32** (3), 155-168 (2020).
4. Castaldo, J., Ochoa, J. Mechanical injury of peripheral nerves. Fine structure and dysfunction. *Clinics in Plastic Surgery.* **11** (1), 9-16 (1984).
5. Singh, A. Extent of impaired axoplasmic transport and neurofilament compaction in traumatically injured axon at various strains and strain rates. *Brain Injury.* **31** (10), 1387-1395 (2017).
6. Singh, A., Kallakuri, S., Chen, C., Cavanaugh, J. M. Structural and functional changes in nerve roots due to tension at various strains and strain rates: an in-vivo study. *Journal of Neurotrauma.* **26** (4), 627-640 (2009).
7. Singh, A., Lu, Y., Chen, C., Kallakuri, S., Cavanaugh, J. M. A new model of traumatic axonal injury to determine the effects of strain and displacement rates. *Stapp Car Crash Journal.* **50**, 601 (2006).
8. Singh, A., Lu, Y., Chen, C., Cavanaugh, J. M. Mechanical properties of spinal nerve roots subjected to tension at different strain rates. *Journal of Biomechanics.* **39** (9), 1669-1676 (2006).
9. Singh, A., Shaji, S., Delivoria-Papadopoulos, M., Balasubramanian, S. Biomechanical responses of neonatal brachial plexus to mechanical stretch. *Journal of Brachial Plexus and Peripheral Nerve Injury.* **13** (01), e8-e14 (2018).
10. Zapałowicz, K., Radek, A. Mechanical properties of the human brachial plexus. *Neurologia I Neurochirurgia Polska.* **34** (6 Suppl), 89-93 (2000).
11. Zapałowicz, K., Radek, A. *Annales Academiae Medicae Stetinensis.* **51** (2), 11-14 (2005).

12. Zapałowicz, K., Radek, M. The distribution of brachial plexus lesions after experimental traction: a cadaveric study. *Journal of Neurosurgery: Spine.* **29** (6), 704-710 (2018).

13. Kawai, H. et al. Stretching of the brachial plexus in rabbits. *Acta Orthopaedica Scandinavica.* **60** (6), 635-638 (1989).

14. Marani, E., Van Leeuwen, J., Spoor, C. The tensile testing machine applied in the study of human nerve rupture: a preliminary study. *Clinical Neurology and Neurosurgery.* **95**, 33-35 (1993).

15. Lee, S. K., Wolfe, S. W. Peripheral nerve injury and repair. *JAAOS-Journal of the American Academy of Orthopaedic Surgeons.* **8** (4), 243-252 (2000).

16. Rickett, T., Connell, S., Bastijanic, J., Hegde, S., Shi, R. Functional and mechanical evaluation of nerve stretch injury. *Journal of Medical Systems.* **35**, 787-793 (2011).

17. Topp, K. S., Boyd, B. S. Structure and biomechanics of peripheral nerves: nerve responses to physical stresses and implications for physical therapist practice. *Physical Therapy.* **86** (1), 92-109 (2006).

18. Lu, Y., Chen, C., Kallakuri, S., Patwardhan, A., Cavanaugh, J. M. Development of an in vivo method to investigate biomechanical and neurophysiological properties of spine facet joint capsules. *European Spine Journal.* **14** (6), 565-572 (2005).

19. Kallakuri, S. et al. Tensile stretching of cervical facet joint capsule and related axonal changes. *European Spine Journal.* **17** (4), 556-563 (2008).

20. Abdel-Aziz, Y. I., Karara, H. M. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. *Photogrammetric Engineering & Remote Sensing.* **81** (2), 103-107 (2015).

21. Pourcelot, P., Audigié, F., Degueurce, C., Geiger, D., Denoix, J. M. A method to synchronise cameras using the direct linear transformation technique. *Journal of Biomechanics.* **33** (12), 1751-1754 (2000).

22. Hedrick, T. L. Software techniques for two- and three-dimensional kinematic measurements of biological and biomimetic systems. *Bioinspiration & Biomimetics.* **3** (3), 034001 (2008).

23. Chen, L., Armstrong, C. W., Raftopoulos, D. D. An investigation on the accuracy of three-dimensional space reconstruction using the direct linear transformation technique. *Journal of Biomechanics.* **27** (4), 493-500 (1994).

24. Singh, A., Magee, R., Balasubramanian, S. Methods for in vivo biomechanical testing on brachial plexus in neonatal piglets. *Journal of Visualized Experiments.* (154), e59860 (2019).

25. Black, J., Ellis, T. Multi camera image tracking. *Image and Vision Computing.* **24** (11), 1256-1267 (2006).

26. Cardenas-Garcia, J. F., Yao, H. G., Zheng, S. 3D reconstruction of objects using stereo imaging. *Optics and Lasers in Engineering.* **22** (3), 193-213 (1995).

27. Mahan, M. A., Yeoh, S., Monson, K., Light, A. Rapid stretch injury to peripheral nerves: biomechanical results. *Neurosurgery.* **85** (1), E137-E144 (2019).

28. Rydevik, B. L. et al. An in vitro mechanical and histological study of acute stretching on rabbit tibial nerve. *Journal of Orthopaedic Research.* **8** (5), 694-701 (1990).

29. Mahan, M. A., Warner, W. S., Yeoh, S., Light, A. Rapid-stretch injury to peripheral nerves: implications from an animal model. *Journal of Neurosurgery.* **133** (5), 1537-1547 (2019).

30. Singh, A., Ferry, D., Balasubramanian, S. Efficacy of clinical simulation based training in biomedical engineering education. *Journal of Biomechanical Engineering.* **141** (12), 121011-121011-121017 (2019).