5-4-2022

Implementation of a Vector Correlation Strain Tracking Algorithm for use with Quantitative Polarized Light Imaging

Ethan Meitz
Washington University in St. Louis

Implementation of a Vector Correlation Strain Tracking Algorithm for use with Quantitative Polarized Light Imaging

Ethan Meitz
Department of Mechanical Engineering and Material Sciences
Washington University in St. Louis
4 May 2022

Abstract:

Full field strain mapping of collagenous tissue typically uses fiducial markers; however, fiducial markers have limited spatial resolution and obstruct light traveling to the sensor when used in conjunction with optical techniques such as Quantitative Polarized Light Imaging (QPLI) [1]. To overcome the limitations of fiducial markers, Quinn et al. proposed a new optical strain tracking method for QPLI that relies on image texture that arises from polarimetry outcomes to calculate strain [1]. Using a modified definition of correlation, image texture generated from QPLI data was tracked between frames to measure deformation. Based on their pseudocode, we implemented the strain mapping algorithm in Python and validated it on a set of control QPLI phantom videos. Future work will extend this algorithm to use of experimental QPLI data; the present state of our implementation struggles to track image texture in videos with low temporal resolution and videos where the signal strength was weak.

Introduction:

Quantitative Polarized Light Imaging (QPLI) uses the polarization state of light to measure fiber alignment in collagenous tissue [2]. By measuring fiber alignment, the progression or onset of fatigue damage can be longitudinally monitored [2]. Information gleaned from this technique can be used to better understand pathological degeneration of musculoskeletal soft tissues and inform rehabilitation techniques [2]. Traditional imaging techniques, like MRI or CT scans, cannot detect fatigue damage of collagen due their poor temporal and spatial resolution [3]. QPLI, which provides a real time quantitative measure of tissue microstructure with high spatiotemporal resolution, is often used as an alternative to these aforementioned techniques.

Traditional QPLI yields the degree of linear polarization (DoLP) and angle of polarization (AoP) which describe the collagen microstructure in the sample [4]. Insights into structure-function relationships in tissues of interest can theoretically be obtained by linking microstructural information obtained with QPLI to local mechanics measurements. By measuring local strain and local microstructure, we hypothesize that we will be able to monitor the onset of mechanically mediated damage substantially earlier than by monitoring catastrophic mechanical failure alone.

Local tissue strain is typically measured via use of fiducial markers. Fiducial markers are physical markings that sit on the tissue sample throughout the test. The markers are typically textured or highly reflective to make optical tracking easy. However, fiducial markings typically yield low resolution strain maps, require tissue preparation, and impede the optical train during imaging [1]. To overcome these issues, Quinn et al. developed an optical strain tracking method that uses DoLP and AoP as image texture to track features through computer vision [1]. Instead of tracking physical fiducial markings, the Quinn et al. algorithm creates virtual markers that can be tracked from frame to frame. This method is extremely promising because it requires no change to the previously established QPLI setup and has good spatial resolution.

Our objective this semester was to implement the texture-based strain tracking algorithm, generate control QPLI phantom videos, and validate the implemented algorithm with these phantoms. Time permitting, we also aimed to extend the algorithm to use with real experimental QPLI data.

Methods:

Strain Map Algorithm Overview:

To use QPLI data as a means of strain tracking Quinn et al. combined QPLI outputs into a vector field that will be used as image "texture" [1]. Once the image texture was generated an initial grid of virtual markers was superimposed onto the vector field. These markers were connected into a mesh using Delaunay triangulation. The local texture surrounding each marker was then used to update a marker's location from frame to frame. Essentially, the algorithm looked for a point in the image where the vector field matches the vectors surrounding the marker. Once all the marker positions for a frame were updated the principal strain for each mesh element was calculated and saved. This process was repeated for every frame in the video or until there were no active markers.

Vector Field & "Texture" Generation

To convert QPLI data into a trackable "texture", the DoLP and AoP values were combined to create a vector field. The DoLP is a value from 0-1 that provides a measurement of collagen fiber strength of alignment and the AoP is an angle values from 0-180˚ that provides information about the orientation of collagen fibers [5]. For each vector, the magnitude was defined as the DoLP, and the angle was set at twice the AoP. Normally, AoP ranges between 0-180˚ so the angle was doubled to span 0-360˚. This vector field will be referred to as an alignment map. The vectors can be decomposed into their x and y components according to Eqn. 1:

$$v_x = DoLP * cos\ (2 * AoP) \qquad v_y = DoLP * \sin\ (2 * AoP) \qquad (1)$$

Vector Map Correlation

The alignment map provided a simple way to combine the outputs of QPLI but created a new challenge when calculating the correlation between two sets of vectors. To update the position of a marker the algorithm looked for the region of vectors that correlated best with the vectors near a marker. Therefore, without a proper correlation definition the tracking algorithm could not work. To calculate correlation each vector was treated as a complex number: $v_x + v_y j$ [1]. The correlation between a set of vectors z and w was calculated using Eqn. 2 [6]:

$$\rho_{zw} = \frac{\sigma_{zw}}{\sigma_z \sigma_w} \quad (2)$$

$$\sigma_{zw} = \frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})^* * (w_i - \bar{w})$$

$$\sigma_z^2 = \frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})^* * (z_i - \bar{z})$$

where $\bar{z}$ is the mean vector, and $z^*$is the complex conjugate.

Tracking Marker Position:

To track a marker from one frame to another, a characteristic sample of vectors called the "fingerprint" is selected on the first frame. The marker (A) and the respective fingerprint surrounding that marker can be seen in Fig. 1. The fingerprint encapsulates the texture around the

chosen marker and can be compared to other regions of the alignment map to track point A in frame N to point B' in frame N+1.

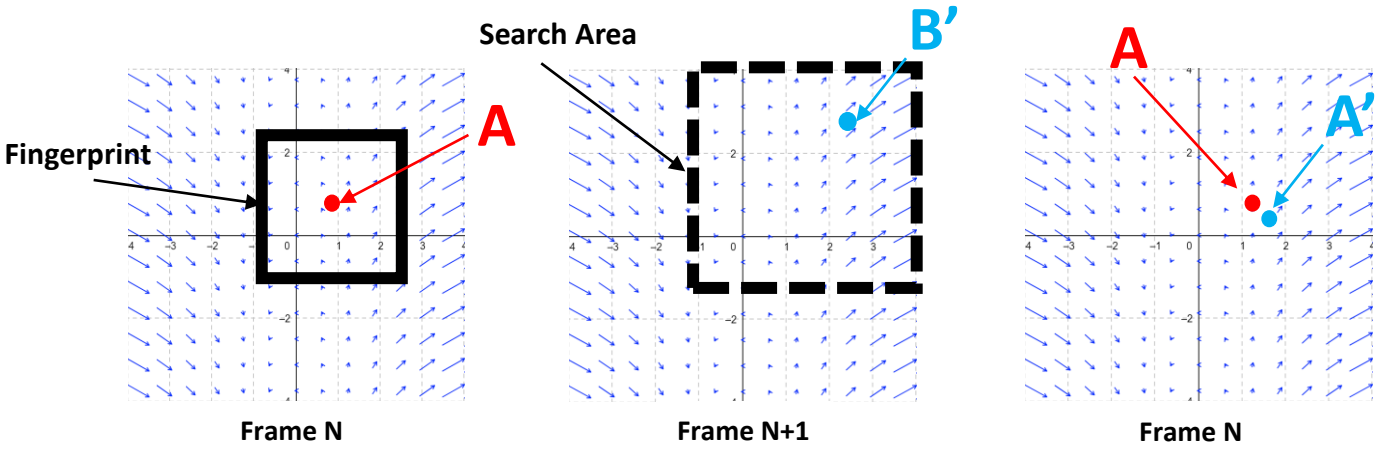


Figure 1: An overview of how marker positions are updated between two adjacent frames

To find B', a search area is centered around A in frame N+1. Then, the fingerprint for point A is compared to each equivalent window in the search area. The point with the highest correlation (Eqn. 2) to the fingerprint is chosen as B'. To improve the accuracy and reduce systematic error, the algorithm is also run in reverse. In pure translation, going from A→B' should be the same as going from B'→A, but with deformation the pixel values do not just translate they also change value. As a result, it is impossible to perfectly backtrack from B' to A because the vectors change position, magnitude, and angle. Therefore, tracking backwards from B'→A yields a new point: A' (**Figure 1**). To reduce error, if the distance between A' and A is greater than a pre-defined threshold, the marker is deactivated. The final step incorporates the information from the backwards time step to define the new marker location, B, on frame N+1 as [1]:

$$B = A + \frac{1}{2}((A - B') - (B' - A'))$$

Finally, if the correlation between fingerprint A and fingerprint B is less than 0.9 the fingerprint for the marker is updated from fingerprint A to fingerprint B. This means that the

fingerprint for a marker only changes once the texture around the marker has appreciably changed. If the algorithm did not update the fingerprint, it would lose its ability to track a feature as the tissue deforms away from the original state.

Strain Calculation:

To calculate strain, a triangular mesh was created with Delaunay triangulation and the Constant Strain Triangle method was used to get the principal strain in each mesh element [7]. The

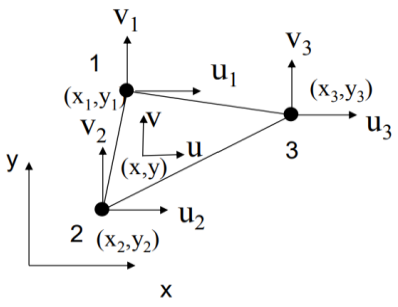


Figure 2: A representative element of the mesh grid.

math to calculate the strain on a representative element (**Figure 2**) of the mesh grid can be found in Appendix A.

Implementation:

The vector correlation algorithm was written in Python using an object-oriented approach and packages available through Anaconda. The algorithm takes two inputs: the spacing between markers and the size of the search area when tracking markers. The search area can be approximated in a displacement control test as a function of the maximum displacement, but the marker spacing was purely guess and check. The program includes a config file so that users can modify the input parameters easily.

Algorithm pseudocode is as follows:

Preprocess data
Generate initial grid of markers
Initialize mesh with Delaunay triangulation
FOR image in data:
    Convert AoP and DoLP to vector field
    PARALLEL FOR *marker in markers:*
        Update marker positions
    Calculate strains

Validation:

To validate the code several control videos were made to test different modes of marker movement. Each control video is represented as a greyscale intensity (**Figure 3**) that scales linearly with the vector magnitude and angle. Therefore, black pixels represent vectors with 0 magnitude and 0 angle and white pixels represent vectors with a magnitude of 1 and angle of 360˚. The first

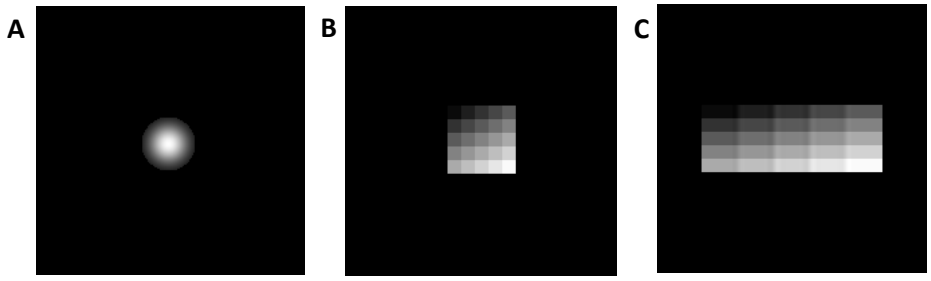


Figure 3: Panel A shows the gaussian bump used in the translation test. Panel B shows the undeformed grid and Panel C shows the grid at the end of the uni-axial deformation test.

control video (**Figure 3A**) was a gaussian bump that translated across the screen. There was no deformation in this video, so the tracking algorithm should work perfectly. The other control videos (**Figure 3B, C**) tested uni-axial deformation, shear, and rotation on a square grid with varying intensity. Each square in the grid represents a set of vectors with the same AoP and DoLP. Therefore, the image "texture" is at the edges of the grid where the vector values transition.

Results:

For each control video 15 frames were generated (0-14). Frame 0 is the initial condition and frame 14 is the most deformed state.

Translation Control:

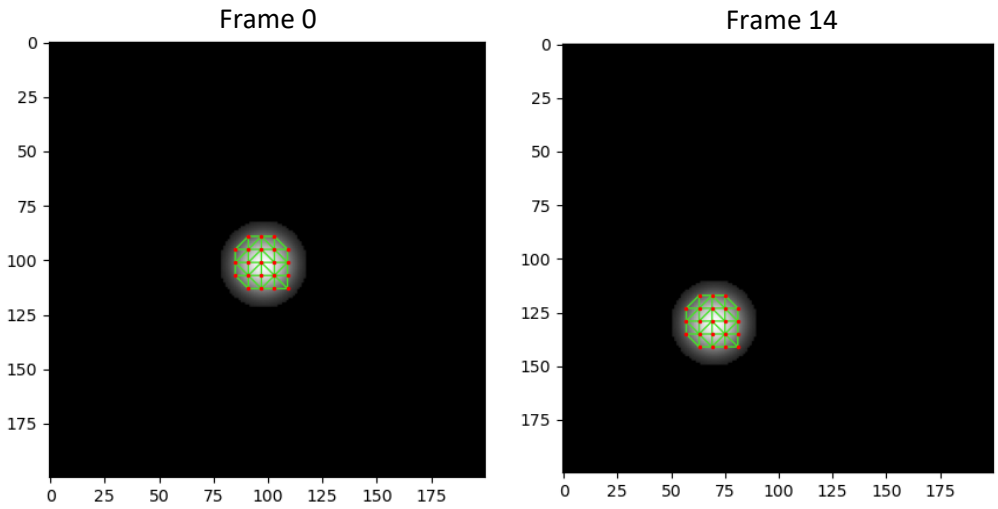


Figure 4: Initial and final frame for the displacement test. The red points are the markers and the green lines show the triangular mesh.

In frame 0 the gaussian bump starts centered in the frame. The circle moves two pixels left and two pixels down in each frame, so in frame 14 the bump has moved approximately 42 pixels. Throughout that displacement the markers retain their original position relative to the bump as it moves around the screen.

Uni-Axial Deformation Control:

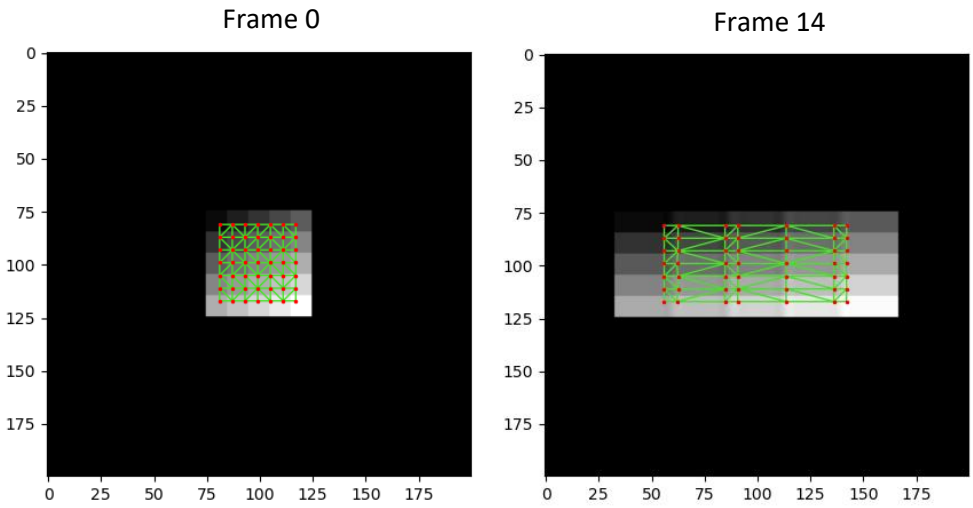


Figure 5: The final and initial frame in the uni-axial displacement test. The final grid is twice as wide as the original.

The markers begin evenly spaced and track the corners of the grid as the square deforms into a rectangle. The deformation was scaled so that the final grid was twice as wide as the original frame ($\varepsilon_x = 1$). The markers do not retain the even spacing, but they do retain their original position relative to the nearest corner of the greyscale grid.

Bi-Axial Deformation Control:

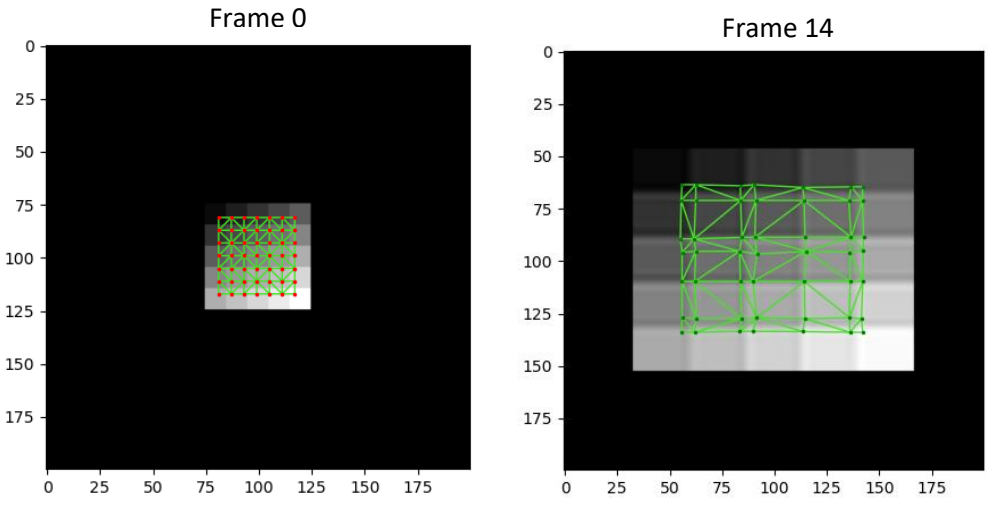


Figure 6: The initial and final frame in the bi-axial deformation control.

The markers behave like the uni-axial control video and track the corners of the grid. Again, the total deformation is twice the original grid ($\varepsilon_x = 1,\ \varepsilon_y = 1$). In this video, the markers begin to drift away from a uniform grid. For example, the four markers near (80, 130) form a parallelogram instead of a square.

Shear Control:

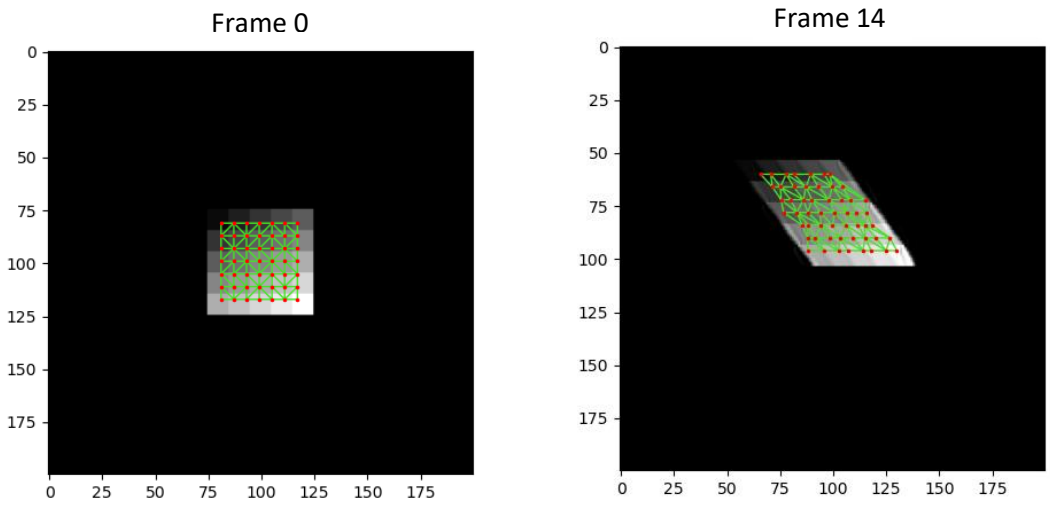


Figure 7: The initial and final frame in the shear control video.

The markers do not retain a perfect grid spacing, but generally follow the corners of the initial grayscale grid. The final shear angle is roughly 22.5˚. The function used to generate shear was not ideal and left artifacts in the image which blurred the transition between squares in the grid. The function also caused the initial condition in frame 0 to displace upwards by about 20 pixels. This displacement was unintended.

Rotation Control:

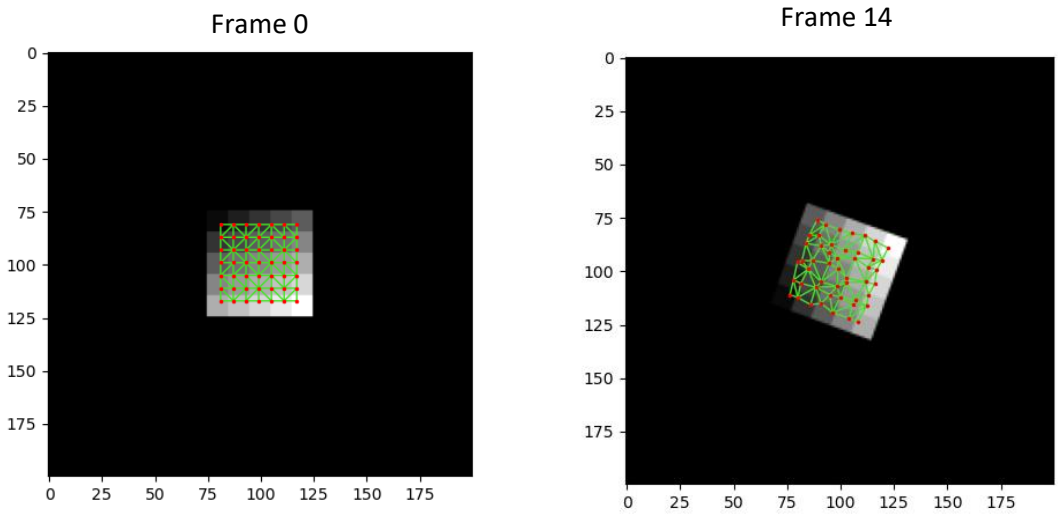


Figure 8: The initial and final frame of the rotation test. The square has rotated 75˚ from its original position.

In the rotation control, the grid was rotated 75˚ around its center. Again, the markers are near their initial positions, but lose the grid structure as the square rotates. The markers on the periphery of the grid appear to retain their positions better than the markers at the center of the grid.

Low Texture Video:

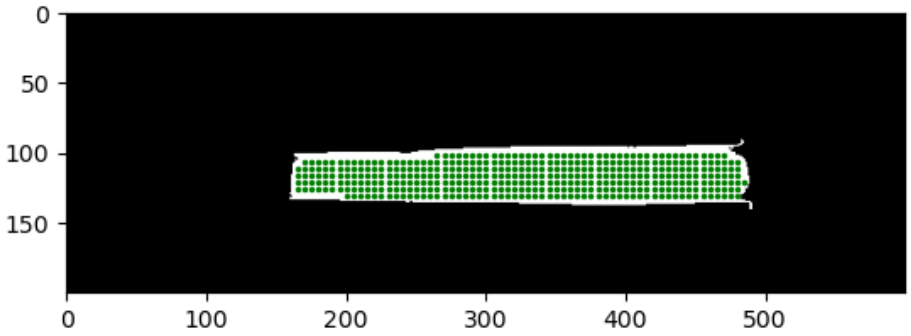


Figure 10: The initial frame of a collagen gel in a uni-axial tension test. The small gaps between markers are an artifact of plotting. Green markers are active.

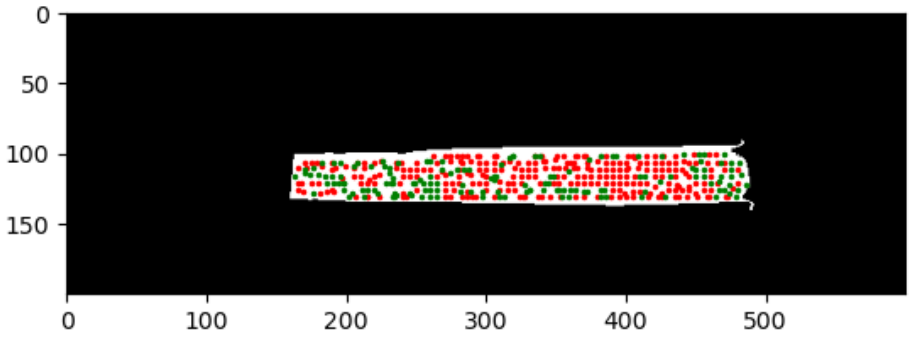


Figure 10: The sixth frame of a collagen uni-axil tension test. The red markers have deactivated to decrease systematic error.

The data used for the low texture test is a collagen gel cyclically loaded in uni-axial tension [2]. The first panel in Fig. 10 shows the initial state of the sample with a grid of markers placed onto that sample. The white mask in the background is the tissue sample and the black region has been masked out so that the background does not interfere. The markers begin green to indicate they are active and turn red when the algorithm deactivates them. After only six frames most of

the markers deactivate to decrease systematic error. The error threshold in this test was increased to five pixels whereas the suggested was two pixels [1].

High Texture Video:

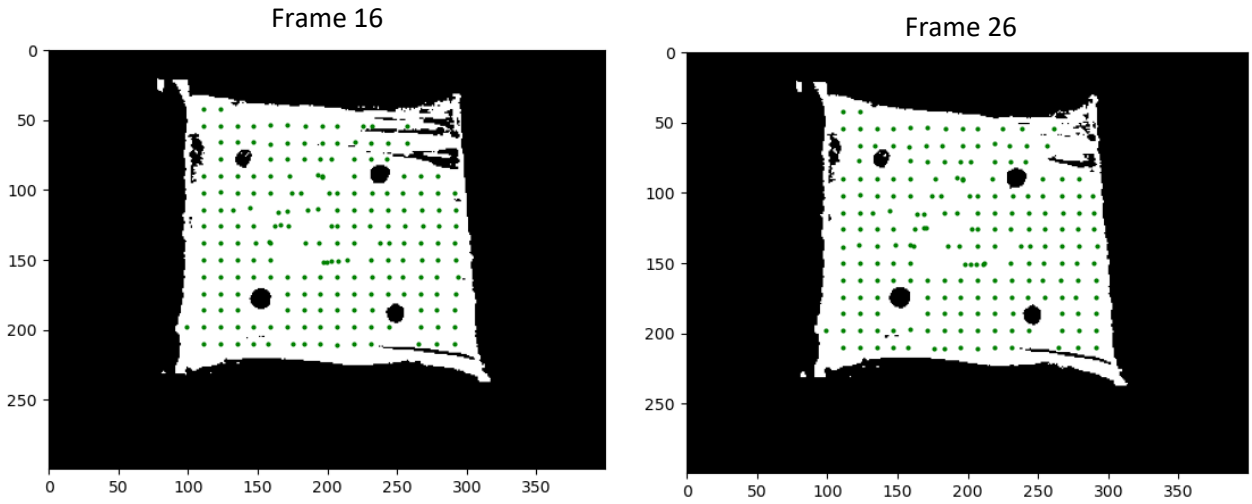


Figure 11: A bovine flexor tendon in uni-axial tension. The four black spots on the tissue are strain beads which block the signal.

Unlike the low texture sample the high texture video is a bovine felxor tendon in a uni-axial tension test. The bovine flexor tendon provides much texture and the markers do not de-active immediately. For the frames between 16 and 26 no marker de-activated. The most clear sign of deformation is in the upper right corner of the tisue where the mask changes from black to white. The tissue is being pulled laterally and contracts perpendicular to the applied load due to the Poission effect. The markers near the deformation do not appear to move, and the markers in the center of the tissue move in an inexplicable way.

Discussion:

This work explained the implementation and validation of a vector-correlation algorithm for strain tracking in QPLI. High resolution strain maps can elucidate the effect of collagen microstructure on fatigue and failure on tissue which cannot be detected with traditional imaging techniques. To validate the algorithm control videos were created that tested different modes of deformation independently. In each control video the tracking algorithm performed as expected. In the translation video, the markers retained their original spacing and followed the gaussian bump around the frame perfectly. The algorithm performed equally as well in the uni-axial deformation test. The markers did not retain uniform spacing, but they did track the corners of the greyscale grid. In the shear and rotation tests, the markers became slightly disorganized. This was because the fingerprint for each marker changed between frames. In translation and uni-axial deformation, the original fingerprint could be found exactly in every following frame. This was not the case in shear and rotation, so the markers drifted slightly from the organized grid. For actual data, rotation equivalent to the control video (75˚) is unlikely. Therefore, the control videos show that the vector correlation code is implemented correctly.

The vector correlation algorithm struggled to accurately track tissue deformation with experimental QPLI data. In a video chosen for its high texture (**Figure 11**), the markers did not track the tissue's motion well. This could be because the film had insufficient temporal resolution. The tracking algorithm relies on the fact that the difference between frames is small. Therefore, if the framerate was too slow the tracking algorithm will fail. In the low texture video (**Figure 10**), the framerate could also be too low, but the markers deactivated after only a few frames. Marker deactivation is a feature of the algorithm designed to reduce systematic error, but it is also a sign of low texture. Both problems could be ameliorated by tuning the fingerprint size or search size; however, changing the input parameters also poses problems. If the search size is too large or the

fingerprint size too small, then the algorithm runs slowly. The code was parallelized to fix this problem, but the runtime still prohibited further analysis of the algorithm with real data in its current state.

Conclusion:

A texture-map based, vector correlation algorithm for strain tracking in QPLI data was successfully implemented. We demonstrated that it could track translation and uni-axial deformation in QPLI phantom videos. The shear looks correct but can only be verified by calculating the shear strain using the triangular mesh. There was not enough time to do this during the semester but will be completed this coming summer. The rotation video cannot be verified as there is no strain applied. Further work also needs to be done to get the algorithm working with real QPLI data. The primary challenge with real data is determining proper input parameters. This takes time and the results are extremely sensitive to the chosen parameters. That said, all the current problems are surmountable given enough time. The vector correlation algorithm works in principle and just needs to be tweaked for this specific use case.

References:

[1] Quinn, Kyle, and Beth Winkelstein. “Full Field Strain Measurements of Collagenous Tissue by Tracking Fiber Alignment through Vector Correlation.” *Journal of Biomechanics*, 2010.

[2] Iannucci, Leanne et al. “Optical Imaging of Dynamic Collagen Processes in Health and Disease”. *Frontiers in Mechanical Engineering.* 2022.

[3] Martin, S. D., et al. “New Technology for Assessing Microstructural Components of Tendons and Ligaments.” *International Orthopaedics*, vol. 27, no. 3, June 2003, pp. 184–89. *DOI.org (Crossref)*, doi:10.1007/s00264-003-0430-4.

[4] Skelley, Nathan W., et al. “Differences in the Microstructural Properties of the Anteromedial and Posterolateral Bundles of the Anterior Cruciate Ligament.” *The American Journal of Sports Medicine*, vol. 43, no. 4, Apr. 2015, pp. 928–36. *PubMed*, doi:10.1177/0363546514566192.

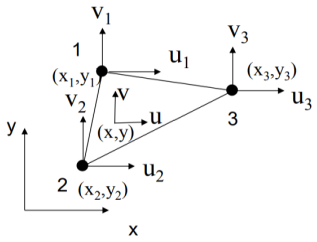
[5] Foster, James J., et al. *Polarization Vision—Overcoming Challenges of Working with a Property of Light We Barely See*. preprint, Animal Behavior and Cognition, 24 Oct. 2017. *DOI.org (Crossref)*, doi:10.1101/207217.

[6] B. Hanson, K. Klink, K. Matsuura, S. M. Robeson, and C. J. Willmott, “Vector correlation: review, exposition, and geographic application”. Annals of the American Association of Geographers. 1992.

[7] De, Suvranu. *Constant Strain Triangle*. Introduction to Finite Elements. Rensselaer Polytechnic Institute.

Appendix A:

Strain Calculation:



Displacement Vector: $\vec{d} =< u_1, v_1, u_2, v_2, u_3, >$

-Displacements are taken relative to initial position so all strains reference the same state

Area of Triangle: A = $\frac{1}{2}\det\begin{bmatrix}1 & x_1 & y_1\\ 1 & x_2 & y_2\\ 1 & x_3 & y_3\end{bmatrix}$

B Matrix: [B] = $\frac{1}{2A}\begin{bmatrix}y_2-y_3 & 0 & y_3-y_1 & 0 & y_1-y_2 & 0\\ 0 & x_3-x_2 & 0 & x_1-x_3 & 0 & x_2-x_1\\ x_3-x_2 & y_2-y_3 & x_1-x_3 & y_3-y_1 & x_2-x_1 & y_1-y_2\end{bmatrix}$

Strain: $\vec{\varepsilon} = \begin{bmatrix}\varepsilon_x\\ \varepsilon_y\\ \varepsilon_{xy}\end{bmatrix} = [B]\,\vec{d}$