2.2 Digital Volume Correlation (DVC)

2.2.1 DVC Approach and Software Overview

The use of digital imaging methods to measure material deformations has become popular because of its non-destructive and scalable approach. The lack of available commercial DVC software packages has the led the author and his colleagues to develop a customized DVC algorithm for 3D displacement tracking and strain quantification. Motivation for the development of the customized DVC algorithm was provided by the MATLAB-based, open-source DIC software package used for 2D displacement tracking developed by Eberl *et al*. [37]. The DVC algorithm employed in this study correlates volumetric images of a sample in its deformed state with a volumetric image of the sample in its undeformed state. Acquiring volumetric images at small, discrete deformation increments allows the DVC algorithm to incrementally track 3D material deformations and compile a cumulative evolution of the applied 3D deformation. In this study, uniaxial mechanical loading is used to apply incremental deformations to the samples.

As described previously, volumetric images are constructed by stacking together sequentially ordered planar image slices acquired at different depths in a sample volume (see Figure 2.9). A piezoelectric objective scanner is used to systematically alter the focal plane depth, and reflectance-based LSCM is employed to obtain high depth-resolution digital images. Each volumetric image contains a unique 3D intensity distribution that is used in the correlation process to determine pattern matching. The volumetric image intensity distributions are measured in voxels (*i.e.*, volumetric pixels). Subvolumes are extracted from each volumetric image (see Figure 2.10), and mathematical correlation is performed on a deformed subvolume with respect to a corresponding undeformed subvolume. Spacing and length-scale parameters within the DVC algorithm are used to control subvolume placement and size. Using anisotropic subvolumes (*e.g.*, 40 x 40 x 20 voxel^3) in the correlation process has proven to be useful given the anisotropic nature of the volumetric images. The subvolumes are centered on gridded control points, which are the locations of where displacements are calculated. The DVC algorithm uses correlation to determine the location of each deformed subvolume with respect to its corresponding undeformed subvolume. The computed shifts between subvolumes provide a means for tracking cumulative 3D displacements, and are used as updated control point inputs for the next iteration of subvolume correlation.

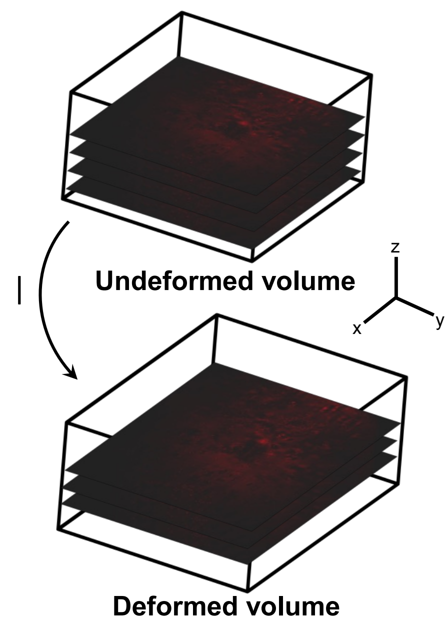

Figure 2.9: Representation of volumetric images in both deformed and undeformed states. Volumetric images are formed by stacking together sequential LSCM image slices at different depths.

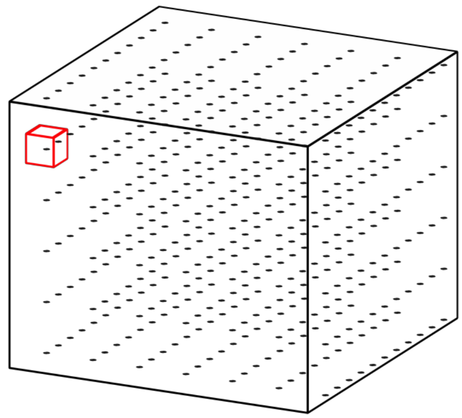

Figure 2.10: Depiction of volumetric image (black) with extracted subvolume (red) centered on gridded control point.

The DVC algorithm uses MATLAB as the engine for performing calculations, and requires seven distinct MATLAB files to provide 3D strain quantification: `filelist_generator3.m`, `grid_generator3.m`, `automate_image3.m`, `cpcorr3.m`, `normxcorr3.m`, `findpeak3.m`, and `displacement3.m`. For a detailed description of the each file see Appendix. The DVC algorithm involves three main steps: pre-processing, correlation, and post-processing. The pre-processing stage consists of collecting a series of undeformed and incrementally deformed volumetric images, storing the filenames of all of the images, and establishing a 3D grid of control points. The correlation stage consists of assembling the individual images into volumetric images, constructing arrays of extracted undeformed and deformed subvolumes, and correlating corresponding deformed and undeformed subvolumes. The post-processing stage consists of displacement data filtering and visualization, as well as displacement field analysis for 3D strain quantification. Figure 2.11 presents a flow chart analysis of the DVC algorithm.

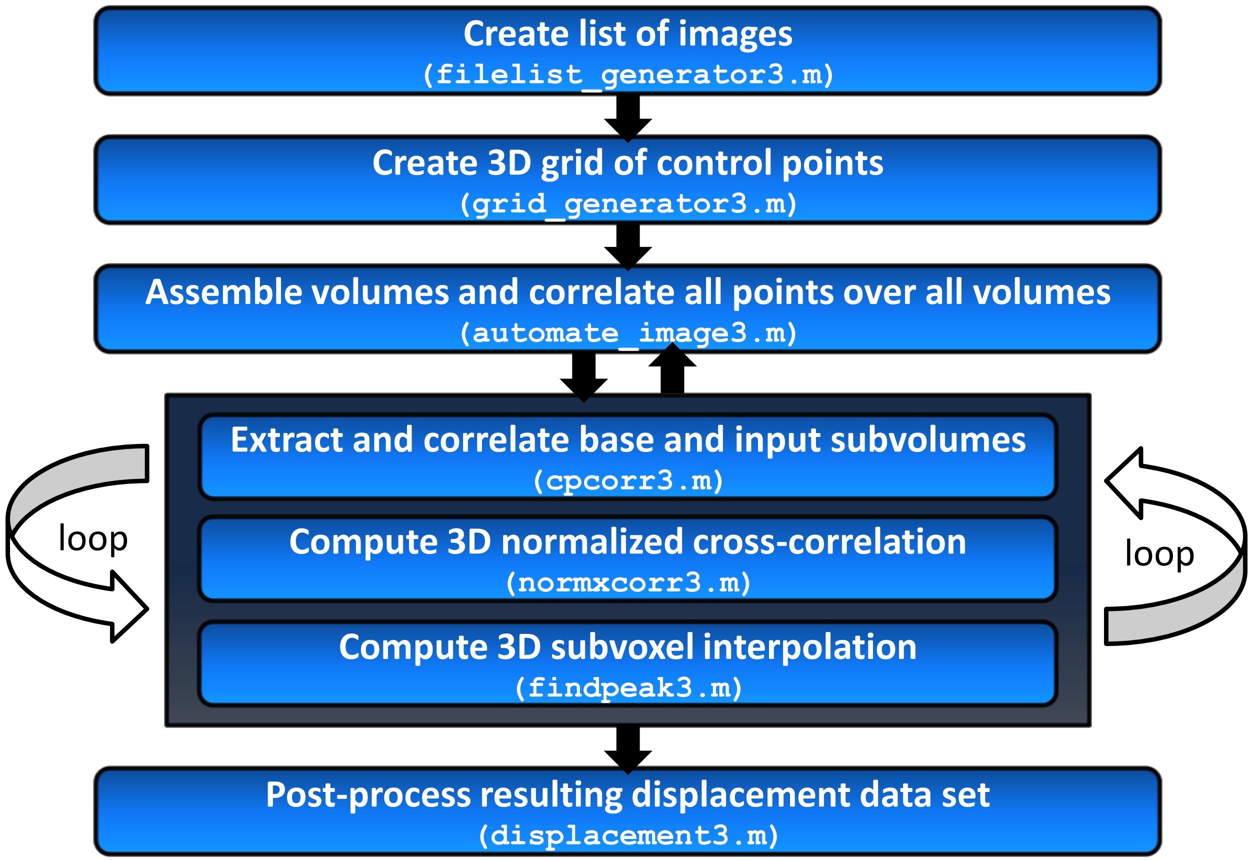

Figure 2.11: Step-by-step process flow of DVC algorithm with associated MATLAB files.

2.2.2 Correlation Algorithm

Calculation of displacement is achieved through normalized cross correlation maximization between two 3D arrays of voxel intensity patterns that represent corresponding undeformed and deformed subvolume pairs. Using a translation-based template pattern matching approach, the general form of the normalized cross correlation coefficient (NCCC), r_{ijk}, can be expressed as follows:

$$r_{ijk}(u,v,w) = \frac{\sum_i \sum_j \sum_k \{F(x_i, y_j, z_k) - \bar{F}\}\{G(x_i^*, y_j^*, z_k^*) - \bar{G}\}}{\sqrt{\sum_i \sum_j \sum_k \left[\{F(x_i, y_j, z_k) - \bar{F}\}^2\right] \sum_i \sum_j \sum_k \left[\{G(x_i^*, y_j^*, z_k^*) - \bar{G}\}^2\right]}}. \quad (5)$$

Here $F(x_i, y_j, z_k)$ represents the voxel intensity value at point (x_i, y_j, z_k) in the undeformed subvolume, $G(x_i^*, y_j^*, z_k^*)$ represents the voxel intensity value at point (x_i^*, y_j^*, z_k^*) in the deformed subvolume, and $\bar{F}$ and $\bar{G}$ represent the mean values of the intensity arrays F and G, respectively. The coordinates (x_i, y_j, z_k) and (x_i^*, y_j^*, z_k^*) are related by the deformation that occurs between undeformed and deformed subvolumes. Assuming rigid body translation only, the relation between (x_i, y_j, z_k) and (x_i^*, y_j^*, z_k^*) is expressed by the following 3D affine transformation:

$$\begin{aligned} x^* &= x + u \\ y^* &= y + v \\ z^* &= z + w. \end{aligned} \quad (6)$$

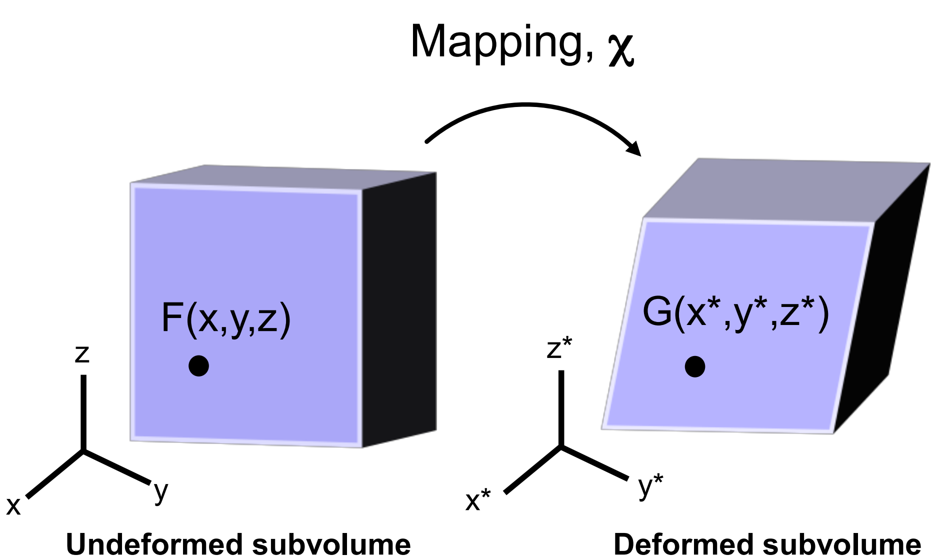

Figure 2.12: Schematic depiction of undeformed and deformed subvolumes with their respective voxel intensities used for evaluation of the NCCC.

The most computationally expensive portion of the DVC algorithm is the calculation of the NCCC array. Improvements in the computational efficiency of the NCCC calculation lead to significant decreases in overall computational time. To this end, the numerator and denominator of Eq. (5) are calculated independently. The numerator of Eq. (5) is defined as the correlation coefficient, R_{ijk}. By setting

$$\begin{aligned} \tilde{F}(x_i, y_j, z_k) &= F(x_i, y_j, z_k) - \bar{F} \\ \tilde{G}(x_i^*, y_j^*, z_k^*) &= G(x_i^*, y_j^*, z_k^*) - \bar{G}, \end{aligned} \tag{7}$$

we can express the numerator of Eq. (5) as follows:

$$R_{ijk}(u, v, w) = \sum_i \sum_j \sum_k \{\tilde{F}(x_i, y_j, z_k)\}\{\tilde{G}(x_i^*, y_j^*, z_k^*)\}. \tag{8}$$

For each trial displacement, (u, v, w), a corresponding correlation coefficient, $R_{ijk}(u, v, w)$, is calculated by summing the products of the intensities at all voxel locations where intensity arrays $\tilde{F}$ and $\tilde{G}$ are coincident. The result generates an array of correlation coefficients, R. By finding the location of the maximum value of R, we can directly determine the corresponding displacement vector that yields highest correlation. To prevent correlation of mean intensities, the mean intensity values, $\bar{F}$ and $\bar{G}$, are subtracted from the subvolume intensity arrays, F and G, prior to correlation.

The greatest improvement in computational efficiency is achieved by calculating cross correlation in the frequency domain. Cross correlation of two functions in the spatial domain is expressed as the product of the first function and complex conjugate of the second function in the frequency domain. Accordingly, the correlation coefficient array can be calculated by taking the Fourier transform of intensity array $\tilde{F}$, the complex conjugate of the Fourier transform of intensity array $\tilde{G}$, and then the inverse Fourier transform of their product. Mathematically, the correlation coefficient array is expressed as follows:

$$R = \mathcal{F}^{-1}\left[\mathcal{F}(\tilde{F})\mathcal{F}(\tilde{G})^{*}\right]. \tag{9}$$

Applying Fourier transforms to compute the cross correlation in the frequency domain and then transforming it back to the spatial domain using an inverse Fourier transform reduces the computation from $\vartheta[N^6]$ operations based on Eq. (5) to $\vartheta[N^3 \log_2 N]$ operations based on Eq. (9) [38],[39].

Furthermore, the denominator of Eq. (5) is calculated efficiently by employing the integral image (summed area table) approach. In 2D, tables containing the integral (running sum) of the image intensity and image intensity square are computed in a single pass over the image. Using the computed summed area tables, the sum of intensities in a rectangular subset of an image are efficiently evaluated simply by knowing the four locations of the rectangular subset corners. For calculation of the denominator of Eq. (5) the integral image approach is extended to 3D, where running sums are efficiently computed over volumetric images instead of planar images.

From Eq. (5), the resulting NCCC array, *r*, contains values ranging from [-1,1]. Perfect positive and negative correlations yield a NCCC of 1 and -1, respectively, while no correlation yields a NCCC of 0. Poorly or spuriously tracked points are discarded by applying a minimum threshold to the NCCC array. A NCCC threshold value of 0.5 is typically implemented to ensure a minimum positive correlation. It should be noted that in the context of the current study, successful DVC "tracking" refers to the correlation of subvolumes with a maximum NCCC above the threshold, and should not be confused with the aforementioned 3D tracking methods of feature particles. The maximization of *r* yields displacement measurements that are accurate to the nearest voxel only. To achieve displacement measurements with subvoxel accuracies, a 3D quadratic polynomial fitting is applied to the maximum and 26 nearest-voxel neighbors of the NCCC array. Solving for the maximum of the 3D quadratic polynomial fit provides a means for interpolating displacements to subvoxel accuracies.