**1. Anisotropy Distribution Histogram**

**Goal: Compute anisotropy (shape factor) for each void in the image.**

**Algorithm:**

1. **Input**: Binary image of voids, or boundary coordinates B{k} for each void.
2. **For each void** k:
   * Extract boundary coordinates:  
     x = B{k}(:,2); y = B{k}(:,1)
   * Center the boundary coordinates:  
     x\_centered = x - mean(x)  
     y\_centered = y - mean(y)
   * Stack as a 2D array:  
     coords = [x\_centered, y\_centered]
   * Compute covariance matrix:  
     C = cov(coords)
   * Compute eigenvalues:  
     [~, D] = eig(C)  
     eigvals = sort(diag(D), 'descend')
   * Compute anisotropy:  
     anisotropy(k) = 1 - (eigvals(2) / eigvals(1))
3. **Output**: Array of anisotropy values for all voids → histogram(anisotropy)

**2. Orientation Distribution (Polar Plot)**

**Goal: Show how major axes of voids are distributed in direction.**

**Algorithm:**

1. **Input**: Eigenvectors of all voids → V\_k, take V\_k(:,1) = major axis.
2. **For each void**:
   * Extract major axis vector → v\_k = V\_k(:,1)
   * Normalize: v\_k = v\_k / norm(v\_k)
   * Store as row in matrix majorEigenVectors
3. **Compute fabric tensor**:
   * fabric = zeros(2)
   * For each vector v in majorEigenVectors:
     + fabric += v \* v' (outer product)
   * Normalize:  
     fabric = fabric / N
4. **Compute deviatoric tensor**:
   * hydro = trace(fabric) / 2
   * dev\_fabric = fabric - hydro \* eye(2)
   * f\_dev = dev\_fabric \* 4 *(Scaling to match 2D fabric literature)*
5. **Polar distribution**:
   * For θ ∈ [0, π]:
     + v = [cos(θ); sin(θ)]
     + radius(θ) = (1/π) \* (1 + v' \* f\_dev \* v)
   * Reflect for [π, 2π]: radius = [radius, radius], theta = [theta, theta + π]
6. **Output**: polarplot(theta, radius)

**3. Anisotropy Norm Evolution (Across Frames)**

**Goal: Track change in bulk anisotropy norm over multiple frames/images.**

**Algorithm (per image):**

1. **Input**: Major eigenvectors for voids in image.
2. **Compute 2D fabric tensor and deviatoric tensor** (same as in plot 2).
3. **Compute anisotropy norm**:
   * norm\_dev = sqrt(2 \* sum(sum(f\_dev .\* f\_dev)))
   * Append to time series:  
     anisotropy\_norm(image\_idx) = norm\_dev
4. **Repeat for each frame**: Loop over image\_idx = 1:n\_frames
5. **Output**:
   * plot(image\_idx, anisotropy\_norm) for time evolution

**Summary of Plots:**

| **Plot** | **X-axis** | **Y-axis** | **Source** |
| --- | --- | --- | --- |
| **Anisotropy Histogram** | Anisotropy bins | PDF | From void-level eigenvalue ratios |
| **Polar Plot** | Orientation angle (θ) | Radius = directional frequency | From major eigenvectors & fabric tensor |
| **Anisotropy Norm Evolution** | Frame index | Frobenius norm of deviatoric tensor | From fabric tensor per image |