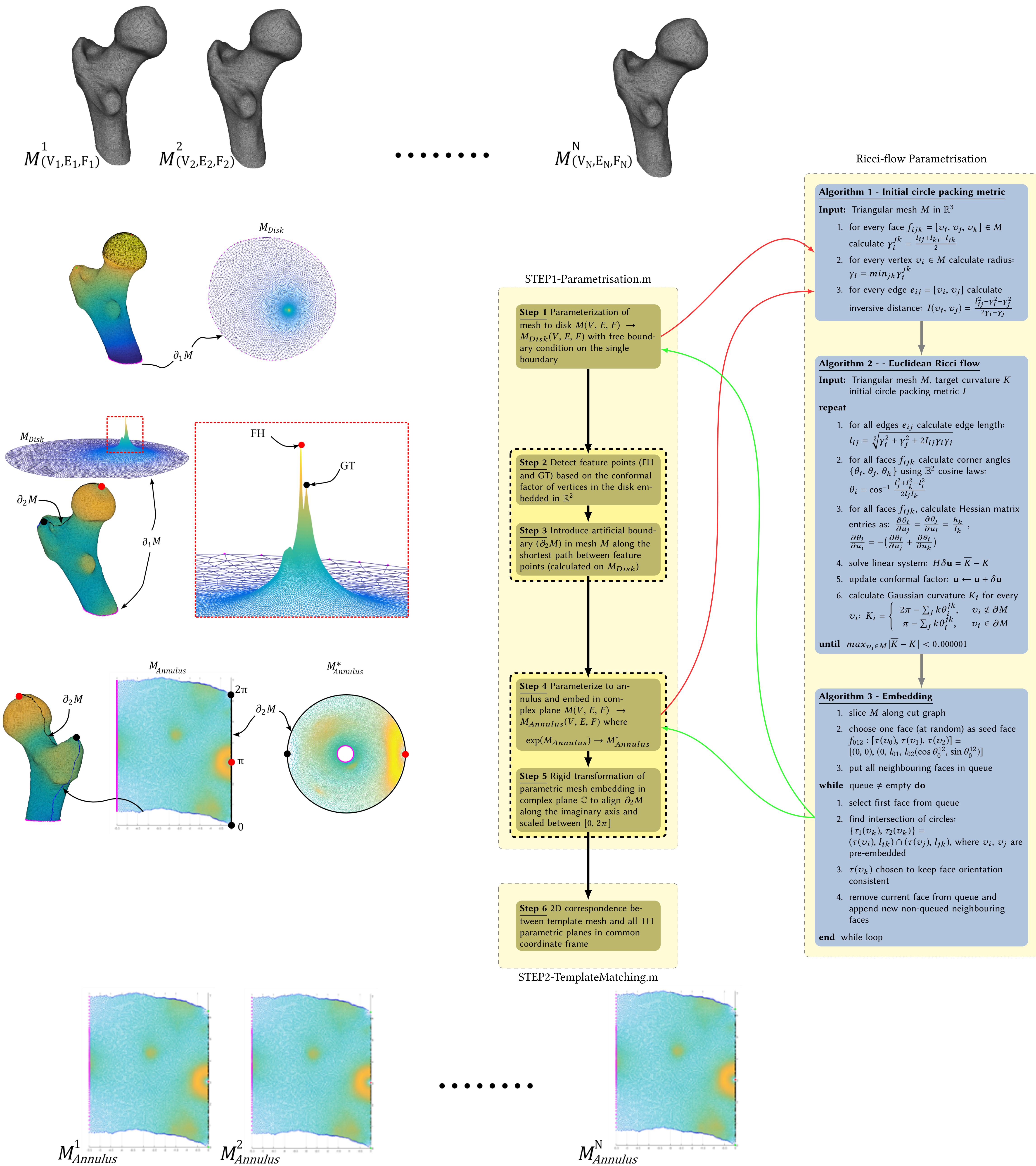

$M^1_{(V_1,E_1,F_1)}$
$M^2_{(V_2,E_2,F_2)}$
$M^N_{(V_N,E_N,F_N)}$
M_{Disk}
$\partial_1 M$
M_{Disk}
FH
GT
$\partial_2 M$
$\partial_1 M$
$M_{Annulus}$
$M^*_{Annulus}$
2π
π
0
$\partial_2 M$
$\partial_2 M$
$M^1_{Annulus}$
$M^2_{Annulus}$
$M^N_{Annulus}$
STEP1-Parametrisation.m
Step 1 Parameterization of mesh to disk $M(V,E,F) \rightarrow M_{Disk}(V,E,F)$ with free boundary condition on the single boundary
Step 2 Detect feature points (FH and GT) based on the conformal factor of vertices in the disk embedded in $\mathbb{R}^2$
Step 3 Introduce artificial boundary ($\partial_2 M$) in mesh M along the shortest path between feature points (calculated on M_{Disk})
Step 4 Parameterize to annulus and embed in complex plane $M(V,E,F) \rightarrow M_{Annulus}(V,E,F)$ where $\exp(M_{Annulus}) \rightarrow M^*_{Annulus}$
Step 5 Rigid transformation of parametric mesh embedding in complex plane $\mathbb{C}$ to align $\partial_2 M$ along the imaginary axis and scaled between $[0, 2\pi]$
Step 6 2D correspondence between template mesh and all 111 parametric planes in common coordinate frame
STEP2-TemplateMatching.m
Ricci-flow Parametrisation
Algorithm 1 - Initial circle packing metric
Input: Triangular mesh M in $\mathbb{R}^3$
1. for every face $f_{ijk} = [v_i, v_j, v_k] \in M$ calculate $\gamma_i^{jk} = \frac{l_{ij}+l_{ki}-l_{jk}}{2}$
2. for every vertex $v_i \in M$ calculate radius: $\gamma_i = min_{jk}\gamma_i^{jk}$
3. for every edge $e_{ij} = [v_i, v_j]$ calculate inversive distance: $I(v_i, v_j) = \frac{l_{ij}^2-\gamma_i^2-\gamma_j^2}{2\gamma_i\gamma_j}$
Algorithm 2 - - Euclidean Ricci flow
Input: Triangular mesh M, target curvature $\bar{K}$ initial circle packing metric I
repeat
1. for all edges e_{ij} calculate edge length: $l_{ij} = \sqrt[2]{\gamma_i^2 + \gamma_j^2 + 2I_{ij}\gamma_i\gamma_j}$
2. for all faces f_{ijk} calculate corner angles $\{\theta_i, \theta_j, \theta_k\}$ using $\mathbb{E}^2$ cosine laws: $\theta_i = \cos^{-1}\frac{l_j^2+l_k^2-l_i^2}{2l_jl_k}$
3. for all faces f_{ijk}, calculate Hessian matrix entries as: $\frac{\partial\theta_i}{\partial u_j} = \frac{\partial\theta_j}{\partial u_i} = \frac{h_k}{l_k}$, $\frac{\partial\theta_i}{\partial u_i} = -(\frac{\partial\theta_i}{\partial u_j} + \frac{\partial\theta_i}{\partial u_k})$
4. solve linear system: $H\delta\mathbf{u} = \bar{K} - K$
5. update conformal factor: $\mathbf{u} \leftarrow \mathbf{u} + \delta\mathbf{u}$
6. calculate Gaussian curvature K_i for every v_i: $K_i = \begin{cases} 2\pi - \sum_j k\theta_i^{jk}, & v_i \notin \partial M \\ \pi - \sum_j k\theta_i^{jk}, & v_i \in \partial M \end{cases}$
until $max_{v_i \in M}|\bar{K} - K| < 0.000001$
Algorithm 3 - Embedding
1. slice M along cut graph
2. choose one face (at random) as seed face $f_{012} : [\tau(v_0), \tau(v_1), \tau(v_2)] \equiv [(0,0), (0, l_{01}, l_{02}(\cos\theta_0^{12}, \sin\theta_0^{12})]$
3. put all neighbouring faces in queue
while queue $\neq$ empty **do**
1. select first face from queue
2. find intersection of circles: $\{\tau_1(v_k), \tau_2(v_k)\} = (\tau(v_i), l_{ik}) \cap (\tau(v_j), l_{jk})$, where v_i, v_j are pre-embedded
3. $\tau(v_k)$ chosen to keep face orientation consistent
4. remove current face from queue and append new non-queued neighbouring faces
end while loop