PS2 — Supplementary Material

CS/ECE 4501/6501/6872: Machine Learning in Image Analysis

Geodesic Shooting for Diffeomorphic Image Registration

$$\frac{dv_t}{dt} = -K[(Dv_t)^T \cdot v_t + (Dv_t) \cdot v_t + v_t \cdot \text{div}(v_t)],$$
$$\frac{d\phi_t}{dt} = v_t \circ \phi_t,$$

where K is a smoothing kernel, D is a Jacobian matrix, $\text{div}(\cdot)$ is a divergence operator, and $\circ$ denotes an interpolation.

→ <u>Jacobian Term</u>

For the vector field, $v = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$ and Dv is a Jacobian matrix.

$$Dv = \begin{bmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} \end{bmatrix} \Longleftrightarrow \begin{bmatrix} v_{xx} & v_{xy} \\ v_{yx} & v_{yy} \end{bmatrix},$$

where, v_{xx} is the gradient of v_x along x-direction, v_{xy} is the gradient of v_x along y-direction and so on.

For each pixel of the image (i, j), the computation of $(Dv_t)^T \cdot v_t$ is:

$$(Dv_t)^T.v_t = \begin{bmatrix} v_{xx}(i,j) & v_{xy}(i,j) \\ v_{yx}(i,j) & v_{yy}(i,j) \end{bmatrix}^T \cdot \begin{bmatrix} v_x(i,j) \\ v_y(i,j) \end{bmatrix}$$

The calculation of $(Dv_t)^T \cdot v_t$ will be a matrix-vector multiplication and the output will be a (2×1) vector.

→ <u>Divergence Term</u>

$\text{div}(v_t)$ is the divergence of the vector velocity field v_t, which produces a scalar at each pixel. For notation simplicity, we omit the time index t below.

$$\text{div}(v) = v_{xx} + v_{yy}$$