# Appendix A

# SPIM Sample Preparation Protocols

## A.1 Nuclear staining of fixed *Drosophila* embryos

### Materials

Dechorionated, fixed and permeabilized embryos stored in methanol at -20°C
PBT(PBS+0.1% Trition)
RNAse solution 100µg/ml
Sytox green dye (Molecular Probes) 5mMolar solution
OR
Sytox green dye (Molecular Probes) 500µMolar stock solution (10µl 5mM solution in 90µl PBT)

### Protocol

1. Move embryos from Methanol to PBT by 4 consecutive 5 minute washes.
   (a) Methanol : PBT 7:3.
   (b) Methanol : PBT 5:5.
   (c) Methanol : PBT 3:7.
   (d) PBT.
2. Prepare RNAse solution (1µl of 100µg/ml in 1000µl PBT).
3. Incubate overnight in RNAse solution.
4. Wash 2×5 minutes in PBT.
5. Prepare final Sytox solution (4µl 500µM stock in 1000µl PBT) (Cover with aluminium foil!).
6. Incubate 1.5 - 2 hours in final Sytox solution (Cover with aluminium foil!).
7. Wash 2×5 minutes in PBT (Cover with aluminium foil!).
8. Mount in agarose for SPIM.

## A.2 Preparation of *Drosophila* embryos for SPIM with Beads

### Materials

Bead stock solution for 40× objective (already prepared or see below how to prepare).
PBT (0.1% Tween or Triton in PBS).
Dechorionated embryos (stained or live).
2% Low melting point (LMP) agarose.
Capillary and plunger for mounting.

### Protocol

Note that if mounting should be done without beads, simply use PBT instead of the PBT/Bead solution.

1. Prepare stock bead stock solution for 40× objective if not available.
   (a) Vortex purchased bead stock at high rate for about 1 minutes (Estapor Microspheres, FXC 050, Conc. 1%, 0.520µm +- 0.037µm) .
   (b) Add 11µl of bead stock to 14 ml of PBT (be very fast after vortexing).
   (c) Vortex final solution, cover with aluminium foil and store in fridge.
2. Take bead stock solution for 40× objective and vortex at high rate for about 1 minute.
3. Depending on the used objective add maybe dilute the solution .
4. Add 500µl to the settled and dried embryos.
5. Gently heat up agarose, ideally in waterbath, otherwise use the microwave (must not boil!).
6. Add 500µl of agarose to the embryos in PBT/Bead solution.
7. Vortex once very gently.
8. Suck embryos into the capillary using the plunger.
9. Try to arrange them in the center of the agarose. Hold the agarose horizontally against the light, let the embryo settle down. Turn the capillary by 180°, wait till the embryos sink approximately to the middle. Then keep rotating the capillary slowly so that they stay in the center till the agarose is solified.
10. Put the capillary in the fridge for 5 minutes, be careful that it does not dry out. Use for example a glass filled with PBS.
11. Mount in SPIM for imaging.

# Appendix B

# Local Coordinate Descriptor Buildup

The *geometric local descriptor* for each detection $\vec{a}$ is defined by its three nearest neighboring detections ($\vec{a}^{n1}$, $\vec{a}^{n2}$, $\vec{a}^{n3}$), to efficiently extract nearest neighbors in image space we employ a 3-dimensional tree. We compute the relative coordinates to the detection which makes each *geometric local descriptor* **translation invariant**.

$$\Delta\vec{a}^{n1} = \vec{a}^{n1} - \vec{a}$$
$$\Delta\vec{a}^{n2} = \vec{a}^{n2} - \vec{a}$$
$$\Delta\vec{a}^{n3} = \vec{a}^{n3} - \vec{a}$$

When matching two views $V_A$ and $V_B$ we have to identify all corresponding *geometric local descriptors* $C_{AB}$ of $V_A$ and $V_B$. Without any optimization we would have to compare all *geometric local descriptors* against each other. To speed up the matching we put a local coordinate system into each *geometric local descriptor* (**Figure** 3.10). The positions of the relative coordinates of the 3 nearest neighbors ($\Delta\vec{a}^{n1}, \Delta\vec{a}^{n2}, \Delta\vec{a}^{n3}$) in the local coordinate system become **rotation invariant** as the local coordinate system is placed into each descriptor in the same way. The coordinate system of a *geometric local descriptor* is built up as described in algorithm 1, the following functions are used

1. $norm(\vec{n})$ normalizes the vector $\vec{n}$ and returns a unit vector pointing in the direction of $\vec{n}$
2. $cross(\vec{n}, \vec{m})$ computes the cross product of $\vec{n}$ and $\vec{m}$ and returns a vector that is perpendicular to $\vec{n}$ and $\vec{m}$
3. $scalar(\vec{n}, \vec{m})$ computes the scalar product of $\vec{n}$ and $\vec{m}$ and returns a scalar
4. $negate(\vec{n})$ negates a vector (inverts the direction)

Now that the local coordinate system is defined by $\vec{x}, \vec{y}, \vec{z}$ we can extract the rotation-invariant scalar values as described in algorithm 2, namely the length of $\Delta\vec{a}^{n3}$ on the x-axis, the x- and y-position of $\Delta\vec{a}^{n1}$ and the x-,y- and z-position of $\Delta\vec{a}^{n2}$ (see also fig. 3.10). We therefore transform

**Algorithm 1** Local Coordinate System Build Up

**Input:** The relative coordinates of the nearest neighbors $\Delta\vec{a}^{n1}, \Delta\vec{a}^{n2}, \Delta\vec{a}^{n3}$,
corresponding to (b),(c),(d) in figure 3.10
**Output:** The vector of the x-Axis $\vec{x}$
The vector of the y-Axis $\vec{y}$
The vector of the z-Axis $\vec{z}$

```
x⃗ ← norm(Δa⃗^n3)                 ▷ The x-axis lies on Δa⃗^n3, with a length of 1.
z⃗ ← cross(Δa⃗^n1, x⃗)              ▷ The normal vector of x⃗ and Δa⃗^n1 will be the z-axis.
z⃗ ← norm(z⃗)                      ▷ The z-axis.
if scalar(z⃗, Δa⃗^n2) < 0 then     ▷ Is z⃗ pointing in the right direction?
    z⃗ ← negate(z⃗)
end if
y⃗ ← cross(z⃗, x⃗)                  ▷ The normal vector of z⃗ and x⃗ will be the y-axis.
y⃗ ← norm(y⃗)                      ▷ The y-axis.
```

those points from the global coordinate system of the view into the local coordinate system of the *geometric local descriptor* where the coordinates can then easily be extracted.

The six scalar values $(v_1 ... v_6)$ can be used to compute the dissimilarity between two descriptors:

$$\varepsilon = (v_1^a - v_1^b)^2 + ... + (v_6^a - v_6^b)^2$$

The lower the difference $\varepsilon$, the more similar they are. To compare all *geometric local descriptor* of $V_a$ against $V_b$ we create a 6-dimensional kd-tree $(v_1 ... v_6)$ for all *geometric local descriptor* of $V_b$, where we query the nearest-neighbor for each *geometric local descriptor* of $V_a$. A *geometric local descriptor*-pair $C_{AB}$ is created if the second-nearest neighbor in the kd-tree of $V_B$ has a significantly higher (10×) $\varepsilon$ than the first nearest-neighbor.

**Algorithm 2** Extract Scalar Values form Local Coordinate System

**Input:** The relative coordinates of the nearest neighbors $\Delta\vec{a}^{n1}, \Delta\vec{a}^{n2}, \Delta\vec{a}^{n3}$, corresponding to (b),(c),(d) in figure 3.10
The vector of the x-Axis $\vec{x}$
The vector of the y-Axis $\vec{y}$
The vector of the z-Axis $\vec{z}$

**Output:** The scalar value $v_1$ (x-position of $\Delta\vec{a}^{n3}$ on the x-axis)
The scalar value $v_2$ (x-position of $\Delta\vec{a}^{n1}$ in the xy-plane)
The scalar value $v_3$ (y-position of $\Delta\vec{a}^{n1}$ on the xy-plane)
The scalar value $v_4$ (x-position of $\Delta\vec{a}^{n2}$ on the xyz-coordinate system)
The scalar value $v_5$ (y-position of $\Delta\vec{a}^{n2}$ on the xyz-coordinate system)
The scalar value $v_6$ (z-position of $\Delta\vec{a}^{n2}$ on the xyz-coordinate system)

1: $v_1 \leftarrow length(\Delta\vec{a}^{n3})$ ▷ v1 is simply the length of $\Delta\vec{a}^{n3}$ as it lies on the x-axis.
2: $\mathbf{T} \leftarrow \begin{pmatrix} \vec{x}_x & \vec{y}_x & \vec{z}_x \\ \vec{x}_y & \vec{y}_y & \vec{z}_y \\ \vec{x}_z & \vec{y}_z & \vec{z}_z \end{pmatrix}$ ▷ Create a 3x3 transformation matrix from the x-, y- and z-axis
3: $\mathbf{T} \leftarrow \mathbf{T}^{-1}$ ▷ Invert the 3x3 matrix
4: $\Delta\vec{a}^{n1} \leftarrow \Delta\vec{a}^{n1} * \mathbf{T}$ ▷ Apply the inverted matrix to $\Delta\vec{a}^{n1}$.
5: $\Delta\vec{a}^{n2} \leftarrow \Delta\vec{a}^{n2} * \mathbf{T}$ ▷ Apply the inverted matrix to $\Delta\vec{a}^{n2}$.
6: $v_2 \leftarrow \Delta\vec{a}^{n1}_x$ ▷ v2 is the x-coordinate of $\Delta\vec{a}^{n1}$.
7: $v_3 \leftarrow \Delta\vec{a}^{n1}_y$ ▷ v3 is the y-coordinate of $\Delta\vec{a}^{n1}$.
8: $v_4 \leftarrow \Delta\vec{a}^{n2}_x$ ▷ v4 is the x-coordinate of $\Delta\vec{a}^{n2}$.
9: $v_5 \leftarrow \Delta\vec{a}^{n2}_y$ ▷ v5 is the y-coordinate of $\Delta\vec{a}^{n2}$.
10: $v_6 \leftarrow \Delta\vec{a}^{n2}_z$ ▷ v6 is the z-coordinate of $\Delta\vec{a}^{n2}$.