Guide to Metrics for Editing Landmarks

Residual Warp Distance - this metric measures the distance between points transformed using the thin-plate spline and the best-fit affine transformations. You start by calculating the best fit affine transformation between the landmarks in LM and EM. This is found by solving for the least squares solution of A in the equation $A\vec{x} = \vec{b}$ where $\vec{x}$ is a vector of the landmarks in the LM space (with each column holding the x, y, and z coordinates for a given point) padded with a row of ones, and $\vec{b}$ is a vector of the landmarks in EM padded with a row of ones. The resulting matrix A will be the best-fit affine matrix for the transformation between the landmarks in LM and EM. Now, given any set of points $\vec{y}$ (including the originals), we can use A to find the affine transformed points $\vec{y'}$. We can also apply the thin-plate spline generated from the landmark file to $\vec{y}$ to generate $\vec{y''}$ (using the ApplyWarpToPoints tool in FIJI or the ApplyBigWarpTrans in Matlab). Letting $\vec{d}$ be the vector given by $||\vec{y'} - \vec{y''}||$, then $\vec{d}$ will be the vector of residual warping distances after applying the best fit affine transformation for each point. Please note that in the Matlab program I deal with the transpose of these vectors for plotting and use by matlab functions, and only use the forms above for the calculations. This has the additional advantage of giving A in the traditional form for an affine transformation matrix.

This metric is a good way to visualize where there is significant warping by plotting a lattice of points as a 3d scatter plot and using a color scheme to indicate the residual warp distance for each point. It can also be used to eliminate points with a residual warp distance over some threshold value, thus helping to eliminate poorly selected landmarks (this can be done using the FindBadWarpPoints function in matlab). However, this metric relies on the underlying assumption that the LM and EM images can be perfectly matched with only translation, rotation, scaling, and shearing, and that any further warping required to match the landmarks is do to errors in the landmark placement. However, we know that the images actually do have some acceptable warping that is due to anatomical variation and errors in the position of the fish when fixing it. Therefore, this metric is very susceptible to false negatives, in which correctly identified landmarks are removed because they are located in areas of the image that have a lot of non-linear warping. In addition, adding in one poorly matched point can change the residual warp distances of all other points. Still, this metric is useful for visualization of the warp field and basic cleaning of points with extreme warping.

Dynamic Residual Warp Distance - this metric uses the same vector $\vec{d}$ as warp distance, but uses a dynamic threshold range rather than a static one when cleaning landmark points. This metric requires a set of carefully selected landmarks $g_{\vec{LM}}$ and $g_{\vec{EM}}$ that are to be considered ground truth in addition to the landmarks $x_{\vec{LM}}$ and $x_{\vec{EM}}$ you are trying to improve. Let $\vec{d_g}$ be the warp distance for the ground truth landmarks file points with the best fit affine matrix A_g generated from the ground truth landmarks, and let $\vec{d_x}$ be the same for $x_{\vec{LM}}$ and $x'_{\vec{LM}}$, where $x'_{\vec{LM}} = A_g x_{\vec{LM}}$. Now we can consider each column of $\vec{d_g}$ to be the “true” amount of non-linear warping at the coordinates for the corresponding column of $g_{\vec{LM}}$. In other words, $\vec{d_g}$ represents the amount of warping that is actually present in the image at each point in $g_{\vec{EM}}$ based off the ground truth landmarks. We can use these points and their corresponding warp distance values to create a linear tetrahedral interpolation[1] $d^* = f(x_i, y_i, z_i)$, where d^* represents our estimation of the true amount of non-linear warping present in the image at the point (x_i, y_i, z_i) (this can be done in Matlab using the CreateWarpInterpolation function). Note that we are assuming all coordinates are in EM space when we calculate the interpolation function, though we could also have used coordinates in LM space as long as we are consistent.

Once we have our interpolation function f, we can use it to evaluate each column of $x_{\vec{EM}}$ to generate a new vector $\vec{d_x}^*$, where $\vec{d_x}^*$ can be thought of as our estimation of the true amount of non-warping present at the image at each point in $x_{\vec{EM}}$. We can now compare $\vec{d_x}$ and $\vec{d_x}^*$ to see how the warping at each landmark pair compares to our estimated true warping at that point in EM. We can then remove all landmarks that result in a warping outside a certain range of our estimated true warping according to the condition $c > \| \vec{d_x}^*{}_i - \vec{d_{x_i}} \|$ where c would be a fixed threshold value.

ANND (Average Nearest Neighbors Distance) - this metric measures the average distance to the 6 nearest points for each point in a point lattice. By measuring the change in this distance for a point lattice before and after it has been transformed by some landmarks file, we can measure the shrinking or stretching around each point in the lattice. The math for this metric is simple, as it is just the average of the euclidean distance between the point of interest and the 6 closest points, which is given by $d_{nn}(P_t) = \frac{1}{6}(\sum_{i=1}^{6} \|P_t - P_i\|)$ for any point P_t. In the point lattice, we know the closest points to any test point P_t will be the two flanking points in the x, y, and z directions. We also know that the distances will all be equal to the distances we set for our point lattice spacing. Therefore, we really only need to calculate this metric for the corresponding points (measured using the same nearest neighbor points as the original, not the actual nearest neighbor points after the transformation) in the lattice after it has undergone the transformation and then compare the two to find the difference.

Please note that as of version 1.0, this metric is no longer supported by this Matlab software package

Localized Residual Warp Distance - to possibly be implemented in a future release, this metric is the same as the other two residual warp distance metrics, accept that the best fit affine transformations are only calculated while only considering some nearest neighbor subset of the points. Therefore, a poorly placed point will only affect the residual warping values of a few local

[1] See “Scattered data interpolation methods for electronic imaging systems: a survey” by Isaac Amidror for details on linear tetrahedral interpolations

points, rather than having a global effect on all points. It would also allow for increased accuracy when removing points by undergoing an iterative process of point removal.