_min]]) * the contact point is the weighted minimum distance ([[#contact_path|_contact]]) \\ ===== background ===== a polygonal surface mesh is defined by a set of faces, each of which is defined by a set of vertices connected by edges. before computing distances, the compute distance map command divides all faces of the rois into triangles, for easier processing and to guarantee that all faces are planar. in the image below, the left image highlights the vertices used to define a surface and the right image displays the triangles that define the surface: compute_distance_map_vertex_triangles.png a normal vector is defined for each triangle, vertex and edge: normals|{{/images/thumb/7/78/compute_distance_map_normals.png ===== command output ===== **** **** |for every result name specified, there will be 6 results:\\ \\ * resultname_segmentname[[#minimum_distance|_min]]\\ * resultname_segmentname[[#contact_path|_contact]]\\ * resultname_segmentname[[#min_vertex_distances|_vert]] | compute_distance_map_data_tree.png| **** **** the result for _min and _contact will have 4 components. the first three components are x y z location of the contact point in the segment coordinate system, and the fourth component is the minimum distance. you can use the [[visual3d:documentation:pipeline:signal_commands:transform_data_coordinate_system|transform_data_coordinate_system]] command to transform the resulting coordinate system of the contact point. ==== minimum distance ==== |the minimum distance between each triangle on one segment's roi and the opposing segment's roi.\\ \\ * a single distance value is calculated for each triangle, and is the smallest value of these candidates:\\ * (a) the distance from the opposing roi to the closest of the triangle's vertices\\ * (b) the distance from the closest vertex on the opposing roi that projects onto the face of the triangle (left image)\\ * (c) the distance from the closest vertex on the opposing roi that projects onto an edge of the triangle (right image) | compute_distance_map_min_distance_calc.png| ==== contact point ==== the weighted minimum distance between the two segments. * once each triangle has been assigned a minimum distance, a preliminary contact point is found by calculating a weighted average of all the triangle centers. each center is weighted by the triangle's area and the square of its distance. the area threshold percentage (contact_point_area_threshold) specifies the percentage of the total area of triangles on the roi to be included in the calculation. this weighted-average point is usually not on the surface of the roi, so it is projected onto the surface using the normal of the closest roi vertex. ==== min vertex distances ==== this is the minimum distance value that is used to color each vertex in the obj. **note:** do not click on this signal and open the data view, visual3d will not be able to open the data view due to the number of columns. ==== sign of the minimum distance ==== any of the distances calculated by this command can be positive (the opposing roi surfaces do not intersect) or negative (the surfaces intersect). the sign of each distance is calculated using the appropriate normal vector (triangle, edge, or vertex). for example, to determine which side of a triangle a given vertex is on, the vector from the vertex to the triangle is compared to the normal of the triangle. ===== command dialog ===== |**result signal information:**\\ \\ * result type: derived\\ * result folder: result can be stored in any folder\\ * result name: one result name for each region you are comparing\\ \\ **color map:**\\ \\ * min color & value\\ * determines the color for the minimum value\\ * all vertices with a minimum distance less than the value specified will have this color\\ * inner color\\ * specifies the mid color and the gradient\\ * max color & value\\ * determines the color for the maximum value\\ * all vertices with a maximum distance greater than the value specified will have this color\\ \\ **event sequence:**\\ \\ * without an event sequence the distance map will be created over all frames of data where the segments exist\\ * when an event sequence is specified the distance map will only be calculated for the specified frames (can be more time efficient to only analyze the time period of interest)\\ \\ **regions:**\\ \\ * segment names: specify the name of the segments to be compared\\ * proximal region & sub region:\\ * the regions correlate to the regions of interest created in the dsx suite\\ * contact point area threshold | distancemapdialog.png| ===== drawing the contact path ===== |you can draw the contact point and its path on the surface: | contactpath.png| ==== steps ====

|for every result name specified, there will be 6 results:\\ \\ * resultname_segmentname[[#contact_path|_contact]]\\ * resultname_segmentname[[#minimum_distance|_min]]\\ * resultname_segmentname[[#min_vertex_distances|_vert]]\\ \\ right click on the _min or _contact to specify the draw properties | compute_distance_map_data_tree_right_click.png| |* target size:\\ * size of contact path (small, medium, large or hide)\\ * note: size of contact path can be modified further by going to\\ * settings -> modify scale of the 3d graphics objects\\ * modify the target size\\ * target color:\\ * specify the color of the contact point\\ * trails:\\ * specify the color for the contact path and the number of frames before/after the current frame to draw the path\\ * trail thickness specifies the thickness of the path | contactpath_draw_properties.png| **note:** if the contact point is still too large, go to **settings -> modify scale of 3d graphics** and decrease the target size. ==== camera settings ==== the skeleton toggle button on the toolbar turns segment drawings on/off: * all on * kinematic on * kinetic on * all off you can set one segment to be always drawn by right click on the segment name under kinetic_kinematic in the data tree, and selecting "draw segment properties". check "draw segment (override visual3d global draw)", this means the segment will always be drawn. you can manually adjust the camera perspective: * right mouse: zoom in/out * left mouse: rotation * right + left mouse: translate you can set the camera perspective to lock onto a segment by going to view -> set camera perspective. }}}}}}}}}}}}}