algorithm by taking advantage of the data provided by our algorithm about the correspondences between the points in the two clouds.

**Figure 7. Coarse Alignment Algorithm.**

1. Points of interest in cloud 1 \( \{p_{1i}\} \) and cloud 2 \( \{p_{2i}\} \) are selected.
2. **while** all the target points \( \{p_{2i}\} \) have not been checked:
3. **while** all the starting points \( \{p_{1i}\} \) have not been checked:
4. for the \( n_e \) resolution levels of the camera images:
5. **Correspondence Search Algorithm** is applied to search in cloud 1 (starting with \( P_{1x} \)) for the correspondence of a point of interest \( P_{2y} \). The output data consist of the corresponding point \( P_{1x} \), the array of cells \( C_{1x} \), the rotation index around the normal, \( k_{tx} \), and the similarity measure, \( M_{Sc} \).
6. if \( M_{Sc} < \tau_{M_{Sc}}(l) \), Go to step 12; else \( \tau_{M_{Sc}}(l) = M_{Sc} \), end if.
7. if level \( l \) is the last one:
   (a) The Euclidean transformation \( T_c \) associated with the correspondence is calculated.
   (b) Two cells of \( C_{1x} \) that meet certain conditions of angle between normals and distance with respect to two other cells in cloud 2 are selected.
   (c) Two correspondences between those cells are established: \( m \leftrightarrow n, r \leftrightarrow s \).
   (d) Using the points \( P_{1x}, P_{1m}, P_{1r} \) from cloud 1 and the points \( P_{2y}, P_{2m}, P_{2s} \) from cloud 2, a fictitious correspondence \( t_1 \leftrightarrow t_2 \) is created and its associated Euclidean transformation is calculated: \( T_f \).
   (e) The distances \( d_R \) and \( d_t \) between the transformations \( T_c \) and \( T_f \) are calculated.
   (f) if \( (d_R < \tau_R) \) and \( (d_t < \tau_t) \), The algorithm is ended. end if
8. end if
9. \( P_{1x} = P_{1x} \) (the first iteration for the next resolution level starts with the point that was found in the previous resolution level).
10. The next resolution level is prepared: \( n_s = 2 \cdot n_s; n_e = n_e/2; k_{tx} = 2 \cdot k_{tx} \).
11. end for
12. \( P_{1x} = \) Next point of \( \{p_{1i}\} \) (if this point has not been evaluated yet)
13. end while
14. \( P_{2y} = \) Next point of \( \{p_{2i}\} \)
15. end while

**Figure 8** Evaluation of the stopping criterion. The points \( p_h \) and \( p_l \) and their normal vectors are obtained from the correspondence \( (p_{1a}, p_{2b}) \) found by the alignment algorithm and two additional correspondences \( (p_{1m}, p_{2m}) \) and \( (p_{1r}, p_{2s}) \). \( p_h \) and \( p_l \) are the centroids of the triangles.