For each voxel in the current keyframe:

* To check if the points in a voxel form a plane, you can use a plane-fitting algorithm. One common method for plane fitting is the least-squares fitting approach. Here are the steps you can follow:
* Compute the centroid of the points in the voxel. The centroid averages the points’ x, y, and z coordinates.
* Compute the covariance matrix of the points in the voxel. The covariance matrix is a matrix that summarizes the variability of the points in the voxel.
* Compute the eigenvectors and eigenvalues of the covariance matrix. The eigenvectors represent the principal axes of the voxel, and the eigenvalues represent the variability of the points along those axes.
* The eigenvector with the smallest eigenvalue corresponds to the normal vector of the plane that best fits the points in the voxel. You can compute the dot product of this normal vector with each point in the voxel to check if they lie on the same plane.
* If the dot product of the normal vector with each point is approximately equal, within a small tolerance, then the points in the voxel form a plane.
* Note that this method assumes that the points in the voxel are noisy and may not exactly lie on a plane. The tolerance used in step 5 should be chosen carefully based on the expected noise in the data.

Using RMS deviation for comparison:

Here are the steps to compute the RMS deviation:

* Compute the plane's normal vector using the covariance matrix's eigenvectors, as described in the previous answer.
* Compute the distance of each point in the voxel from the plane using the equation: distance = abs((point - centroid) · normal). where "point" is a point in the voxel, "centroid" is the centroid of the points in the voxel, "normal" is the normal vector of the plane, and "·" denotes the dot product.
* Compute the RMS deviation using the formula: RMS deviation = sqrt(sum(distance^2) / N). where "distance" is the distance of each point from the plane, "N" is the number of points in the voxel, and "sum" denotes the sum of the squared distances.
* Use the RMS deviation as the tolerance for checking if the points in the voxel form a plane. If the distance of each point from the plane is less than or equal to the RMS deviation, then the points are considered to lie on the plane.
* Note that the RMS deviation provides a measure of the variation of the points from the plane, which can be used to determine a suitable tolerance. If the points are relatively tightly clustered around the plane, a smaller tolerance can be used, whereas a larger tolerance may be needed if the points are more spread out.

Frame overlap calculation:

Here's a possible implementation using MAD:

* Transform the centroids and normal vectors of the planes in frame A to the coordinate system of frame B using the rigid body transformation between the two frames.
* For each plane in frame A, find the closest plane in frame B by comparing their normal vectors. You can do this by computing the dot product between the normal vectors of the planes and selecting the plane with the highest dot product.
* Calculate the distance between the centroids of the matching planes in frame A and frame B.
* Compute the median of the distances between the matching planes.
* Compute the median absolute deviation (MAD) of the distances. MAD is calculated as the median of the absolute deviations of the distances from the median value.
* Set the threshold to a multiple of the MAD. For example, you could use a threshold of 2 or 3 times the MAD to capture a certain percentage of the matching planes.
* Filter out the matching planes that have a distance greater than the threshold.
* To calculate how much the frames overlap, you can compute the area of the overlapping planes. This can be done by intersecting the two planes and calculating the intersection area.
* Repeat steps 2-8 for all planes in frame A and compute the total overlap area by summing the areas of the overlapping planes.