360° Camera Alignment via Segmentation (Supplementary Material)

Benjamin Davidson, Mohsan S. Alvi, and João F. Henriques

Disperse.io
{ben, mohsan, joao}@disperse.io

A Appendix A

A.1 Processing Equirectangular Segmentations to Vertical Directions

To lift our equirectangular segmentation to a uniformly distributed spherical segmentation, we project a subset of points in the equirectangular segmentation to the sphere. This subset is constructed by uniformly distributing n_{sp} points on the surface of the upper hemisphere of the sphere using [1], and also taking the antipodal point to each such point. Sampling the antipodal points like this, allows the assignment of a score to potential vertical axes easily, by summing the probability of a point with its antipodal counterpart. Let v be one of these points on the sphere. To assign a probability to v, it is projected to the output binary segmentation via $p^{-1}(f^{-1}(v))$, and the interpolated probability from pixel segmentation probabilities is assigned. Sampling the equirectangular segmentation uniformly like this gives equal weight to all directions. This is not the case if the equirectangular segmentation is used directly, as undue weight would be placed on the poles due to over sampling.

Given our uniformly distributed spherical segmentation, the objective is to pick a single vertical direction. To do this, the probabilities for all antipodal pairs are summed, and the resulting score is assigned to the point in the upper hemisphere, and all points in the lower hemisphere are discarded. This is subject to our assumption that the vertical direction is in the upper hemisphere. This limits us to correcting images for which the misalignment is less than 90 degrees. In practice this is not a limitation, as images are typically not taken upside down, and if they are, the onboard camera sensors can roughly correct the image before applying our method. Next, we discard all points with a score less than λ_{score}. To extract the vertical direction from the remaining points, first all vectors within a distance λ_{dist} of each other are grouped together. For each such group, we calculate the mean score and mean vector. The vertical direction is then the mean vector with maximal mean score.

A.2 Uniformly distributing the Vertical Axis

We generate n_{rot} rotations which place the vertical direction uniformly on the surface of the sphere. To this end, n_{rot} points are generated uniformly around the sphere using [1]. A rotation is then constructed which places the vertical direction at each of these points. Let v be one such point, then the rotation which transforms z to v is given by the Rodrigues vector:

$$(z \times v)(z). \tag{1}$$

After applying this rotation to z, a small random rotation is applied. By rotating in this manner we ensure that our network will see rotations covering the whole sphere allowing it to generalise well. Moreover, by randomly perturbing these rotations we avoid overfitting to the n_{rot} rotations transforming z to v, as every rotation is slightly different.

References

1. Saff, E.B., Kuijlaars, A.B.J.: Distributing many points on a sphere. The Mathematical Intelligencer **19**(1), 5–11 (1997). https://doi.org/10.1007/BF03024331, https://doi.org/10.1007/BF03024331