## Visual SFM-Based Measure

This visual measure computes egomotion given any two images with at least eight matching features using a state-of-the-art Structure From Motion (SFM) algorithm and then compares the resulting camera poses to the provided hypothesis trajectory. Given intrinsic calibration parameters, the algorithm finds point correspondence between pairs of images, computes the essential matrix, decomposes it into camera rotation and translation, and then triangulates each set of point correspondences in order to obtain the 3D structure of the scene.

Given two sets of matched interest points between the images taken at two nodes  and, the epipolar geometry constraint can be defined as

,

where is the calibration matrix containing the camera intrinsic parameters, and  is the essential matrix. We use the eight point algorithm [1] to obtain a linear estimation of the elements of the essential matrix by rearranging the epipolar equation in the following form:

,

where is the vector made up of the entries of in row-major order, and is formed using the image coordinates of each matched point  as



Each matching pair gives rise to one equation of the form (3); therefore, the null space of the matrix which has at least 8 pairs stacked in its rows gives the linear least squared estimate for. Consequently, the rank constraint of the essential matrix (is of rank 2) can be enforced by

,

where and are the orthonormal matrices obtained from the singular value decomposition of the estimated essential matrix (i.e). Consequently, the projection matrices corresponding to the two views at the two nodes and  can be formed by decomposition of the essential matrix





where is the last column of, and

.

Since there are four possible choices for the second camera matrix, we employ the visibility constraint to select the correct solution. In that, we triangulate each matched pair using each of the solutions, and accordingly select the solution which results in a positive depth for the highest number of pairs. Finally, the estimated cameras are optimized using sparse bundle adjustment [2].

The estimated cameras contain the relative rotation matrixrepresented as quaternion, and the relative translation. Therefore, the recovered motion can be directly compared to the candidate trajectory which is represented as a relative quaternion and a relative translation. Hence, we compute the residual of as

 ,

## Modeling the Residual

In order to obtain a better data-driven residual representation, the residual  resulting from the discussed measures is compared to its characteristic distribution as determined from training data. For instance, we have modeled the tracking error in the Middleburry dataset using a single Gaussian, a Gaussian Mixture Model, and Kernel Density Estimation (KDE). In order to select the most appropriate model, we divided our dataset into two sets for training and testing. The training data is used in the model learning, and then the model response against the testing data is compared to the actual groundtruth response using the sum of squared error (SSE). Figure 1 illustrates the obtained results. In our experiments, we found that even as simple as a single Gaussian is appropriate for modeling the error, and that the SSE decreases with increasing the number of Gaussian components, while being generally close to the SSE obtained when using the KDE. In our framework, we use a single Gaussian component in order to avoid over-fitting. Given the residual  for each feature pair and the error model (eg.), we compute the normalized Sum of Squared Differences (SSD)

.

As discussed in the previous section, follows a chi-square distribution; therefore, the final data likelihood is given as

.

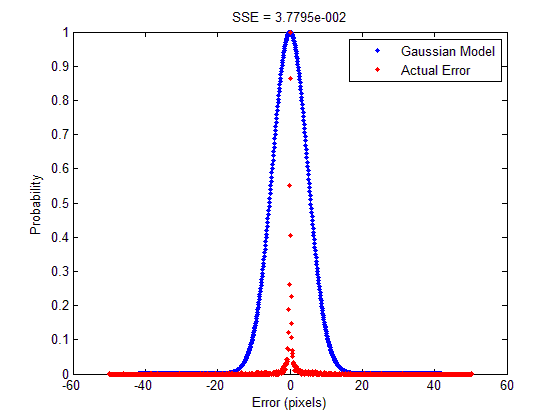


Figure 1. KLT feature tracking error distribution given Middlebury Art data. The blue curve represents the learned model, and the red dots represent the actual data.

References:

[1] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 19(6):580 - 593, October 1997.

[2] M.I.A. Lourakis and A.A. Argyros, “SBA: A Software Package for Generic Sparse Bundle Adjustment”, ACM Transactions on Mathematical Software, vol. 36, no. 1, pp. 1-30 (Mar. 2009).