1. Visual Measures. In this section, we describe two visual measures and investigate the properties of each.

**1-Motion Estimation-Based Measure:**

The ego motion between any two nodes of our graph can be estimated using Structure from Motion (SFM). Given two sets of matched interest points between the images taken at two nodes  a and , the epipolar geometry constraint can be defined as

…(1)

(i.e,), 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:

 …(3)

where is formed using the  and coordinates of the points

.

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 on by

 …(4)

where and are obtained from the singular value decomposition of 

Once has been computed, the projection matrices corresponding to the two views at the two nodes and  can be formed by decomposition of the essential matrix

**…(6)

…(7)

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. Additionally, we apply sparse bundle adjustment [2] as a final step to optimize the obtained cameras.

The recovered motion can be directly compared to the candidate trajectory which is represented as a rotation matrixand a translational vector. Hence, we extract Euler angles , and then compute the normalized residual of as

**….**(9)

.

.

**2-Fast Point-Based Measure:**

Alternatively, the matched sets of points  can also be directly used to compute a residual without the need for computing scene structure. Each feature match is returned as a pair of ray vectors and  in the camera frame, derived from the camera projection. Given a trajectory hypothesis, vectors in the camera frame can be rotated into the world frame as follows:

…..(10)

The two ray vectors in the world frame are then compared to the direction of translation given by the hypothesis trajectory as follows:

….(11)

An alternative way to interpret the previous measure can be expressed in terms of optical flow. In that, we use the hypothesis translation  and rotation  to simulate a translational optical flow field, and a rotational optical flow field  respectively. Consequently, the matched points are represented as flow vectors (note that can also be obtained by regular optical flow estimation methods). The rotational field is then subtracted from the optical flow to obtain a rotation-free flow. The residual of the hypothesis can then be computed as

…(12)

where  is the vector orthogonal to . It can be shown that the residual in equation 13 is equivalent to 11.

**1.1. Modeling the Residual**

In order to obtain a probabilistic model of the residual,, the characteristic distribution of the residual is computed 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 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 error model(eg.), the residual is represented as , which is consequently used to compute cost as in equation (10\*).

\*from the full paper document

TODO: Fix equation numbers

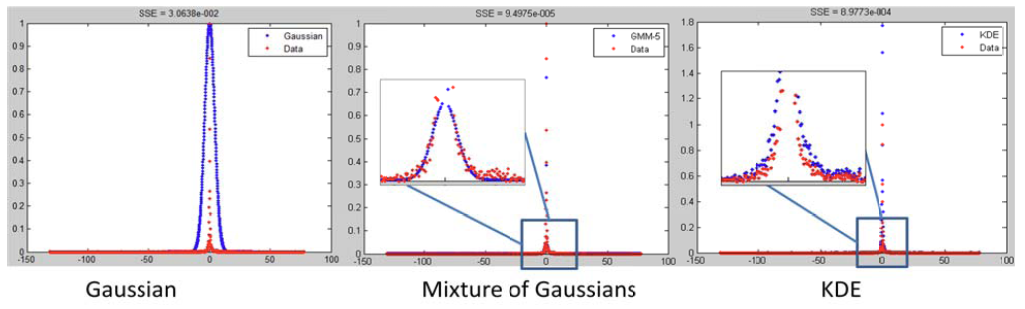


Figure 1. KLT feature tracking error distribution given Middlebury Art data. The blue curves represent the learned model, and the red dots represent the actual data.

**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).