$k - means$. Next the initial estimate of D is computed using 15 while fixing $\tilde{H}$ to it's initial value. This is followed by $\tilde{H}$ computation using 16 while keeping D fixed at it's previously computed value. This process of alternating minimization is repeated till either the change in J falls below a certain threshold ϵ or for a preset number of iterations. We set $\epsilon = 10^{-3}$ and the threshold to 600 in our experiments.

Speedup. Our overall method will be expensive, if the base segmentations are expensive to compute. To remove this bottleneck we use fast minimum variance quantization to produce the base segmentations. The RGB color space is split into a user specified number of levels such that each level minimizes the variance of its constituent pixel values. Thus the obtained segmentation accounts for both color and textural variation in a naive fashion. We further find that not imposing the smoothness constraints further improves performance. This is primarily because computing the constraint matrix Θ proves to be expensive. In our experiments we use an ensemble of 4 base segmentations obtained by using two different color levels $\{6, 10\}$ at different degrees of smoothing.

4. Experiments and Results

We compare consensus segmentation against two other widely used segmentation algorithms, the efficient graph based segmentation (GBIS) algorithm [5] and the mean shift algorithm [4]. GBIS treats the image as a graph. Segmentation is achieved by splitting the graph into a collection of connected components. Two connected components are merged when the weight of the edge connecting the two components is less than the maximum weight in either components' minimum spanning tree, plus some constant user controlled parameter M. The publicly available implementation [5] of this algorithm has two other tunable parameters, σ a smoothing parameter and min the minimum number of pixels in segment. In this paper we fix $\sigma = 0.8$ and vary M through {100,200,300,400,500} and min through {20,50,100,300,500}.

The Mean Shift (MS) algorithm involves a mean shift filtering of the image data followed by a clustering of the filtered data. The mean shift filtering is a search for modes of the underlying pdf of the image data. In this paper we have used the open source EDISON [4] implementation of the mean shift segmentation. The EDISON system converts the original RGB image into the LUV space. The mean shift filtering is carried out in a 5 dimensional feature space, containing the (x, y) image coordinates and the LUV values. The algorithm has three tunable parameters spatial bandwidth (h_s), color bandwidth (h_r) and min. We, following popular trend [24] set $h_s = 7$ and vary h_r through $\{3, 5, 7, 9, 11, 13, 15\}$ and the range of min is chosen to be the same as GBIS.

Our algorithms *consensus* and sped-up consensus *fastCon* have two tunable parameters. We apply a Gaussian filter on our images, and the standard deviation of the filter σ is the first parameter, while the number of color clusters C present in an image is the second parameter. σ is varied through {0.75,1.0,1.75} for *consensus* and through {2,4} for *fastCon*, while C varies from image to image. In our experiments C took an integer value between 3 and 12 depending on the image.

For quantifying the performance of the segmentation algorithms, we use the Probabilistic Rand Index (PRI) proposed in [23]. We compare an image segmentation S^{test} with a set of "ground truth" human segmentations $\{\mathcal{H}^1, \mathcal{H}^2, ..., \mathcal{H}^{\mathbb{H}}\}$. The human segmentations are obtained from the Berkeley Image Segmentation Dataset [14], which contains a test set of 100 images.

For completeness we briefly describe the computation of PRI. A segmentation is considered "good" if it agrees with the human segmentations provided. The PRI score increases if the labels l_i and l_j of two pixels i and j are the same, i.e. if they are classified in the same segment of S^{test}, and they are also classified in the same segment for a human segmentation $\mathcal{H}^h$. The score is hurt if this is not the case. Formally, PRI is computed as:

$$\begin{aligned} \mathrm{PRI}(S^{test}, \{\mathcal{H}^1, ..., \mathcal{H}^{\mathbb{H}}\}) \quad = \quad & \frac{1}{\binom{N}{2}} \sum_{i<j} [\mathcal{I}(l_i^{S^{test}} = l_j^{S^{test}}) p_{ij} \\ & + \mathcal{I}(l_i^{S^{test}} \neq l_j^{S^{test}})(1 - p_{ij})] \end{aligned} \tag{19}$$

Where, $\mathcal{I}$ is the identity function and

$$p_{ij} = \frac{1}{\mathbb{H}} \sum_{i<j} [\mathcal{I}(l_i^{S_k} = l_j^{S_k})] \tag{20}$$

PRI takes values in the range $[0, 1]$, with a value of 0 resulting when S^{test} and $\{\mathcal{H}^1, ..., \mathcal{H}^{\mathbb{H}}\}$ have no similarities, and a value of 1 when S^{test} matches $\{\mathcal{H}^1, ..., \mathcal{H}^{\mathbb{H}}\}$ exactly. The PRI values obtained for the three segmentation algorithms as well as the best $k - means$ base segmentation are presented in Table 1. Figure 3 displays segmentations of a representative subset of the Berkeley test images, for visual comparison. The consensus algorithm significantly outperforms the best base segmentation,thus providing a quantitative measure of utility of the consensus process. Consensus also performs comparably with both MS and GBIS. Predictably (Figure 3), Consensus performs well when regions of the image can be distinguished on the basis of color, texture or some combination of the two. The sped up version *fastCon* produces cheap and competitive but somewhat worse segmentations. On a Intel core 2 duo machine with 4GB of RAM *fastCon* required 0.5 ± 0.015 seconds to execute. The consensus took 0.2365 ± 0.02 seconds while