Supplemental Document for View-consistent 4D Light Field Superpixel Segmentation

Numair Khan Qian Zhang Lucas Kasser Henry Stone *Min H. Kim James Tompkin
Brown University, USA *KAIST, South Korea

1. Expanded Results

To evaluate our results qualitatively, we refer the reader to our supplemental video which demonstrates view consistency by animating between the views. In this document, we present additional information and experimental results to better characterize the performance of the tested methods:

Section 2 An explanation of boundary accuracy metric computations as is common for 2D superpixel methods.

Section 3.1 More qualitative results for central-view superpixel boundaries across synthetic HCI and real-world EPFL Lytro Illum datasets.

Section 3.2 Visualizations of the *average number of labels per pixel* metric, which show the spatial distribution of errors in correspondence by label reprojection into the central view.

Section 3.3 Visualizations of the *achievable segmentation accuracy* metric, which shows the spatial distribution of boundary errors.

Section 3.4 Parameter variation of weight of CIELAB components vs. spatial and depth terms in k-means clustering. We show that increasing intensity and color weight increases boundary performance but decreases compactness and view consistency performance.

Section 3.5 Per dataset metric scores for the HCI dataset, rather than the average presented in the main paper. These show the relative characteristics of the different scenes.

Finally, our open source code and precomputed results are available at https://github.com/brownvc/lightfieldsuperpixels.

2. Superpixel Evaluation Metrics

Along with the two light field specific view consistency metrics used to evaluate our work (*self-similarity* and *number of labels per pixel*, as explained in the main paper), we use four metrics from existing 2D segmentation and superpixel literature to assess non-view specific qualities of our method: *achievable segmentation accuracy*, *boundary recall*, *undersegmentation error*, and *compactness*.

Achievable Segmentation Accuracy Given a ground-truth object-level segmentation map, this metric measures the achievable accuracy possible by the oversegmentation, *e.g.*, in a later interactive object selection stage. To compute the metric, we assign labels to superpixels according to the ground truth object labels from the synthetic HCI dataset [7]. The label which maximizes overlap with a superpixel across views is the assigned label.

Boundary Recall Given a ground truth boundary image G and an algorithm's boundary image B, we compute the fraction R of ground truth edges that fall within a certain distance d of at least one superpixel boundary [3]. We use $d=2$ chessboard distance. True Positives (TP) is the number of boundary pixels in G for whose exist a boundary pixel in B in range d; False Negative (FN) is the number of boundary pixels which do not fall within this range. Boundary recall is:

$$R = \frac{TP}{TP+FN}. \tag{1}$$

Intuitively, the higher the boundary recall, the better the superpixels adhere to object boundaries. However, using this alone favors segments with long boundaries. Thus, we plot the metric with different superpixels sizes and accompany this with the undersegmentation error for more considered evaluation.

Undersegmentation Error A segment S in the ground truth segmentation image G divides a superpixel P into an *in* and an *out* part. The undersegmentation error compares segment areas and provides the percentage of superpixels which overlap ground-truth segment borders. Various implementations of the undersegmentation error metric exist; we adopt the formulation from Neubert et al. [3] which does not penalize large superpixels that have only a small overlap with the ground truth segment:

$$UE = \frac{1}{N_S}\sum_{S\in G}\frac{\sum_{P:P\cap S\neq\varnothing} min(|P_{in}|,|P_{out}|)}{|S|}. \tag{2}$$

The inner sum is the error introduced by this specific combination of ground truth segment and superpixel, depending on their overlap.

Compactness Compactness provides a measure of superpixel boundary curvature. We use Schick *et al.*'s compactness metric [4]:

$$C(\mathcal{S}) = \sum_{S\in\mathcal{S}}\frac{4\pi A_S|S|}{|I|L_S^2}, \tag{3}$$

where $\mathcal{S}$ is the set of superpixels, $|I|$ is the size of a single light field view, and A_S and L_S are the area and perimeter of superpixel S, respectively. We compute the median superpixel compactness per light field for a fixed superpixel size of 20, then average across the HCI dataset.

3. Additional Experiments

3.1. Qualitative Per-View Superpixel Boundaries

We present a single (central) view of the qualitative results for inspection of the superpixel boundaries. First, we note that rating the shape of a superpixel may require knowing what application is in mind; typically boundary recall can be improved if shape 'regularity' is sacrificed, but regular shapes may provide a more consistent expectation for applications with user interaction. Clustering-based methods can pick points within this trade-off by varying the weight of feature terms; we demonstrate this in Figure 6. Second, we note that these results only express boundary shape in a single view, and that view inconsistency can appear as boundary shape error between views. Please refer to our supplemental video for this behavior.

We show full and window cut-out superpixel results for our method, for LFSP [8], and for our baseline method of k-means clustering on a central view depth map computed by the method of Wang et al. [5, 6]. For full details of the baseline k-means method, we refer the reader to Section 4.1 of the main paper. This baseline is also related to the recent work of Hog et al. [2] on light field superpixels for video, which uses SLIC [1] (k-means on a regular initialization grid, as per our method) with angular coordinates as features (cf. our depth feature).

Figure 1 shows superpixel boundaries for the HCI dataset across the 'buddha', 'papillon', 'still life', and 'horses' datasets. Figures 2 and 3 shows boundaries for six images from the real-world EPFL dataset captured with a Lytro Illum camera.

3.2. Spatial Maps for Average Number of Labels per Pixel

Figure 4 shows the average number of labels per pixel metric as a heatmap for the central view, where blue is low (good view consistency) and red is high (bad view consistency). We see that most errors occur at edges; that LFSP has more inconsistency around edges; and that k-means is sometimes sensitive to high-frequency pattern textures.

While this could be alleviated with clustering feature weight parameter tuning, we note that our method does not suffer this issue even though it uses the same component weight parameters as our k-means baseline. This is because we cluster on EPI segments, as outlined in section 3.2, rather than individual pixels. Our method, effectively, performs a per-scanline segmentation before clustering.

3.3. Spatial Maps for Achievable Accuracy

Figure 5 shows a visualization of achievable accuracy via the ground truth semantic-level segmentation maps provided in the HCI dataset. Red denotes where a superpixel crosses a boundary in the ground-truth map. Generally, our approach performs better than LFSP and comparably to our k-means baseline.

3.4. K-Means Parameter Variation

The baseline k-means-based segmentation method clusters each pixel in the central view based on a vector $f = (x, y, d, L^*, a^*, b^*)$ of spatial, depth, and CIELAB color features. For our evaluation, each feature was assigned the same weight parameters as used in the spatio-angular segmentation stage of our method. Figure 6b explores the effect of varying the CIELAB color weight with respect to the spatial and depth parameter. We observe that while increasing the color weight improves performance on 2D superpixel accuracy metrics, performance on the compactness measure and the light field specific metrics, namely self-similarity and average labels per pixels, degrades.

3.5. HCI Per-Scene Quantitative Metrics

For completeness, we include the per-scene qualitative measures on the HCI dataset across Figures 7 to 12. We can see varying complexity across the datasets, e.g., papillon has relatively easier boundaries, while horses has difficult boundary segmentation with text in the background. Different techniques also perform better or worse on different datasets, e.g., our approach does well on still life, but less well on papillon due to our weaker regularization for smooth untextured regions.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels. page 15, 2010. 2

[2] Matthieu Hog, Neus Sabater, and Christine Guillemot. Dynamic super-rays for efficient light field video processing. In *BMVC*, 2018. 2

[3] Peer Neubert and Peter Protzel. Superpixel benchmark and comparison. In *Proc. Forum Bildverarbeitung*, volume 6, 2012. 1

[4] Alexander Schick, Mika Fischer, and Rainer Stiefelhagen. Measuring and evaluating the compactness of superpixels. In *ICPR*, 2012. 1

[5] Ting-Chun Wang, Alexei A Efros, and Ravi Ramamoorthi. Occlusion-aware depth estimation using light-field cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3487–3495, 2015. 2, 3, 4, 5

[6] Ting-Chun Wang, Alexei A Efros, and Ravi Ramamoorthi. Depth estimation with occlusion modeling using light-field cameras. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2170–2181, 2016. 2, 3, 4, 5

[7] Sven Wanner, Stephan Meister, and Bastian Goldluecke. Datasets and benchmarks for densely sampled 4d light fields. In *VMV*, pages 225–226. Citeseer, 2013. 1, 3

[8] Hao Zhu, Qi Zhang, and Qing Wang. 4D light field superpixel and segmentation. In *IEEE CVPR*, 2017. 2, 3, 4, 5

[9] Martin Řeřábek and Touradj Ebrahimi. New light field image dataset. In *Proceedings of the 8th International Conference on Quality of Multimedia Experience (QoMEX)*, 2016. 4, 5

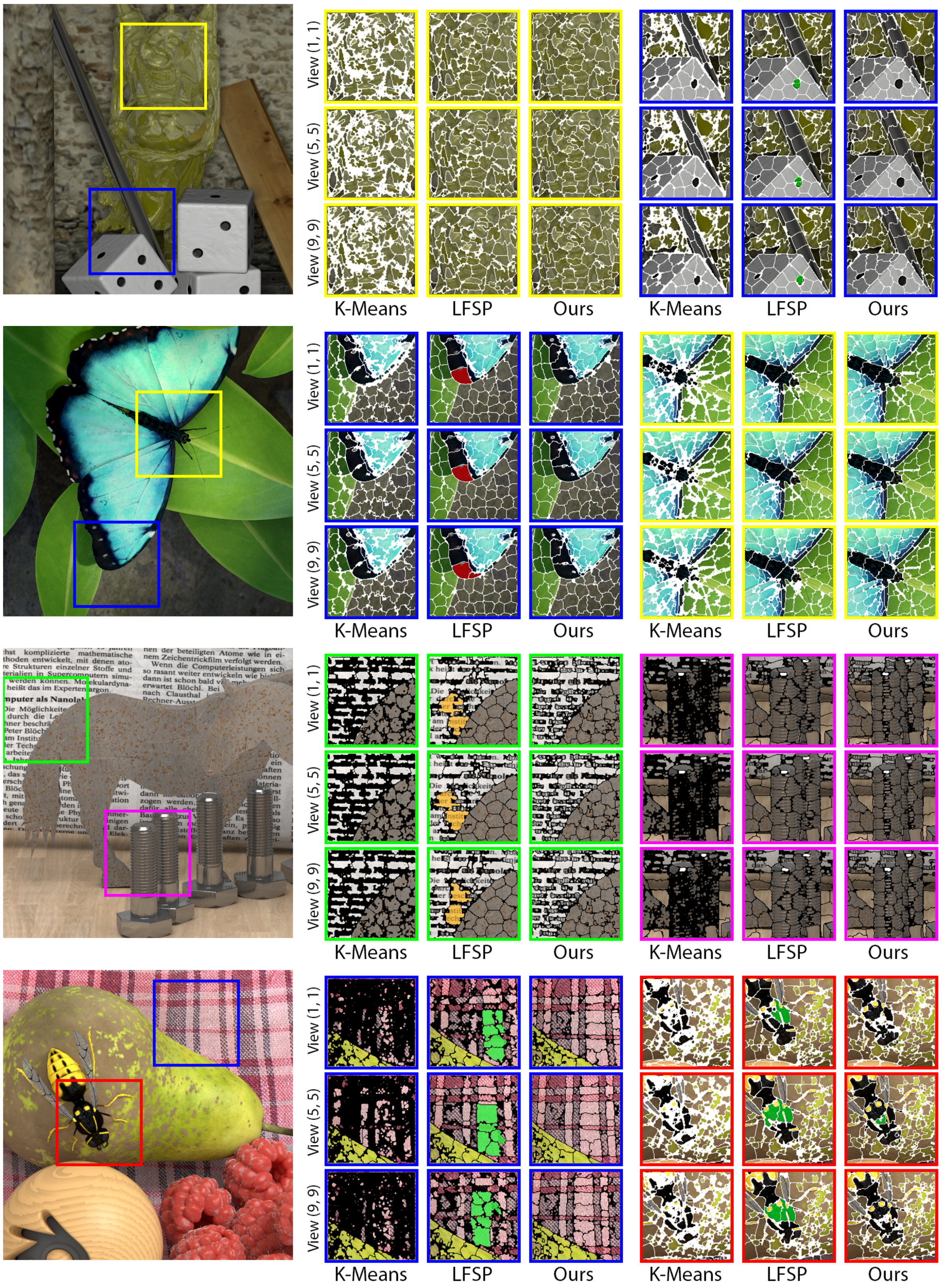


Figure 1: Superpixel segmentation boundaries and view consistency for the k-means baseline, LFSP [8], and our method. Disparity maps for LFSP and k-means were calculated using the algorithm of Wang *et al*. [5, 6]. We include all four HCI dataset [7] light fields for completeness; we highlight superpixels which either change shape or vanish completely across views. Our superpixels tend to remain more consistent over view space, which can be easily seen as reduced flickering in our supplementary video. *Note:* Small solid white/black regions appear when superpixels are enveloped by the boundary rendering width. k-means tends to have more of these regions which helps it increase boundary recall, but this behavior is not useful for a superpixel segmentation method.

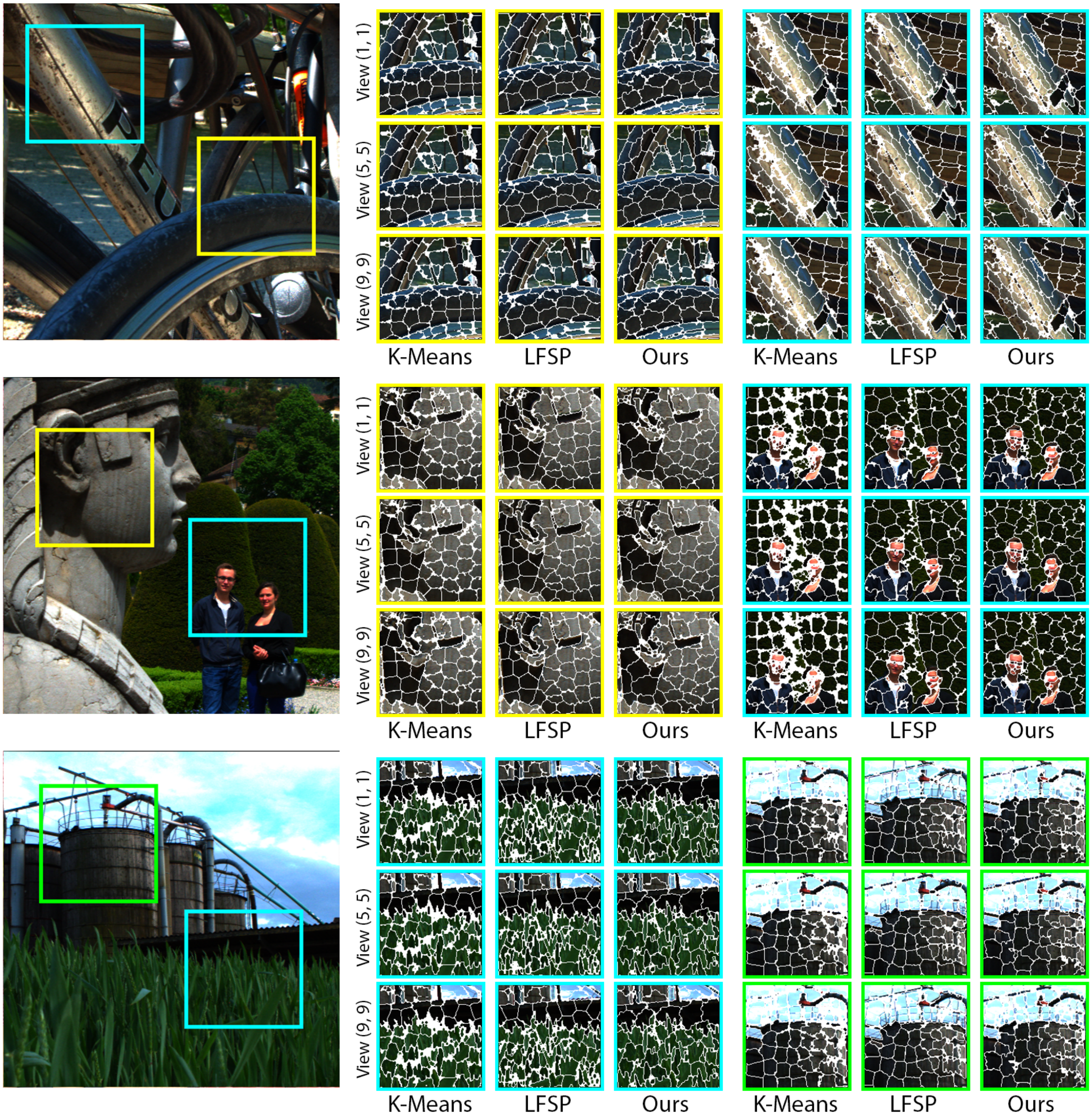


Figure 2: Superpixel segmentation boundaries and view consistency for the k-means baseline, LFSP [8], and our method. Disparity maps for LFSP and k-means were calculated using the algorithm of Wang *et al.* [5, 6]. We include six light fields from the EPFL Lytro light field dataset [9], with three more in Figure 3. Our superpixels tend to remain more consistent over view space, which can be easily seen as reduced flickering in our supplementary video. *Note:* Small solid white regions appear when superpixels are enveloped by the boundary rendering width. k-means tends to have more of these regions which helps it increase boundary recall, but this behavior is not useful for a superpixel segmentation method.

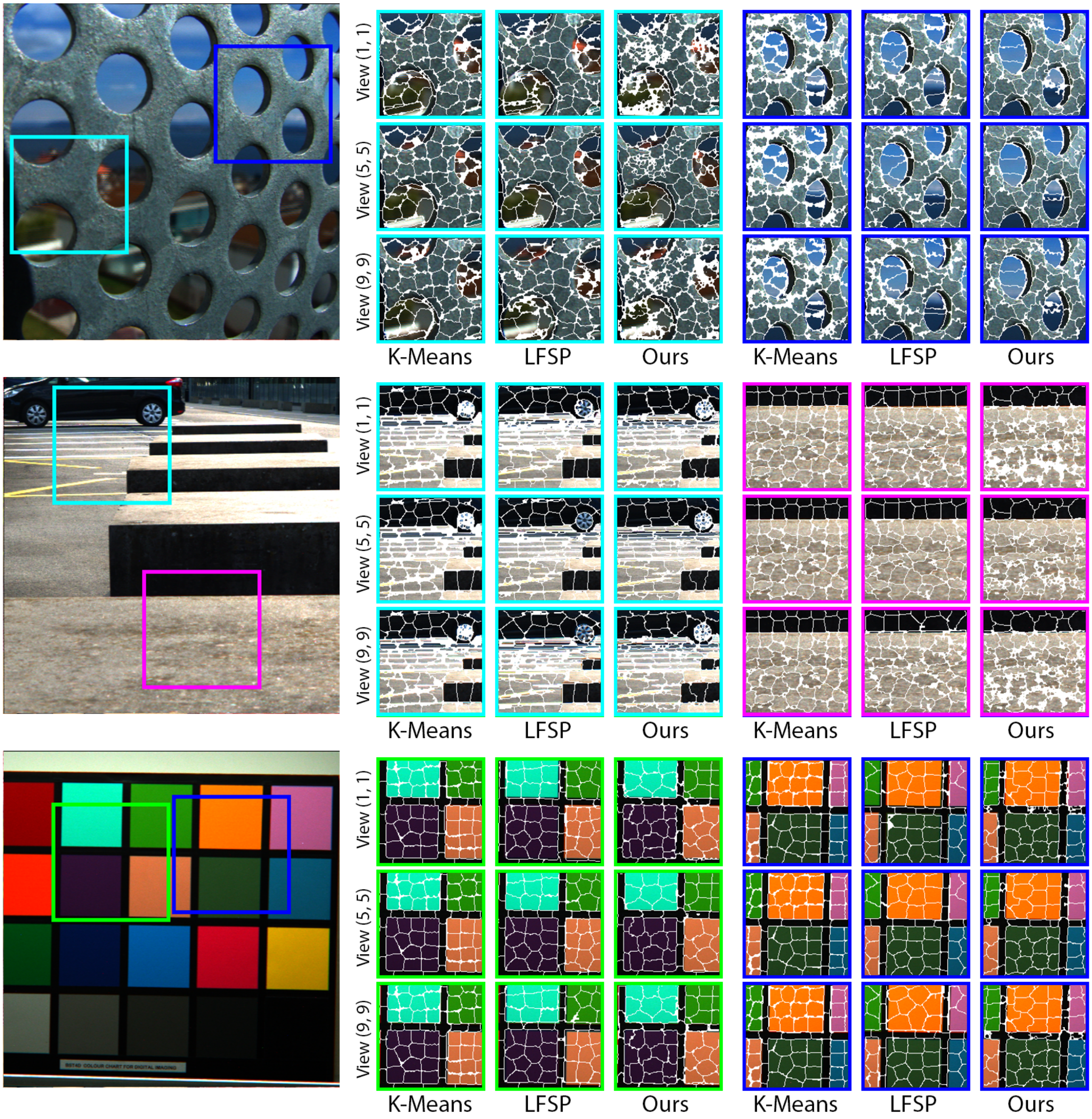


Figure 3: Superpixel segmentation boundaries and view consistency for the k-means baseline, LFSP [8], and our method. Disparity maps for LFSP and k-means were calculated using the algorithm of Wang *et al.* [5, 6]. We include six light fields from the EPFL Lytro light field dataset [9], with three more in Figure 2. Our superpixels tend to remain more consistent over view space, which can be easily seen as reduced flickering in our supplementary video. *Note:* Small solid white regions appear when superpixels are enveloped by the boundary rendering width. k-means tends to have more of these regions which helps it increase boundary recall, but this behavior is not useful for a superpixel segmentation method.

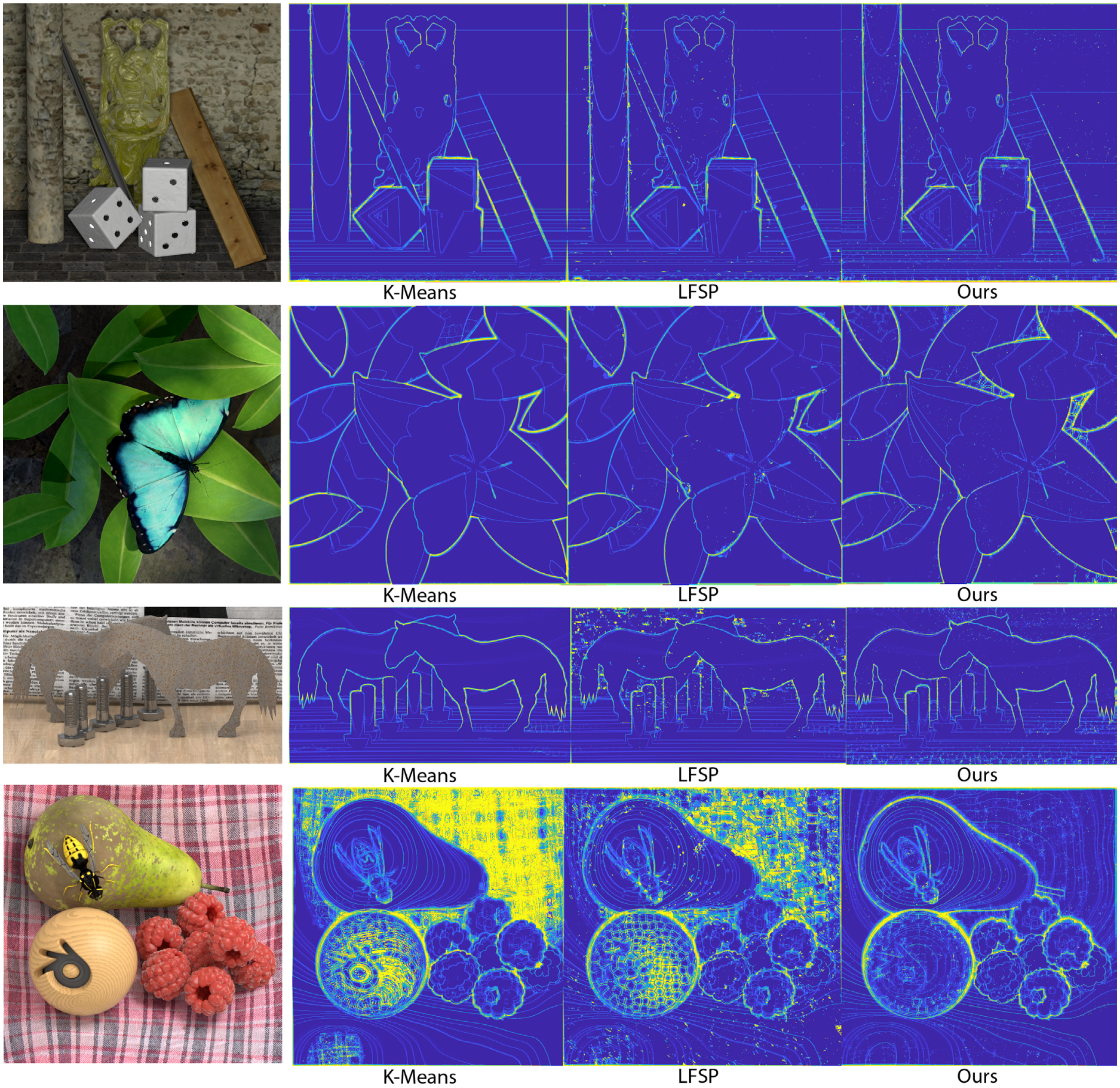


Figure 4: Average number of labels per pixel metric, shown as a heatmap for the central view where blue is low (good view consistency) and red is high (bad view consistency). From left to right, the images are: input, k-means, LFSP, and our result. Both k-means and LFSP use the Wang et al. generated depth map. We can see that most errors occur at edges; that LFSP has more edge inconsistency; and that k-means is sometimes sensitive to high-frequency pattern textures.

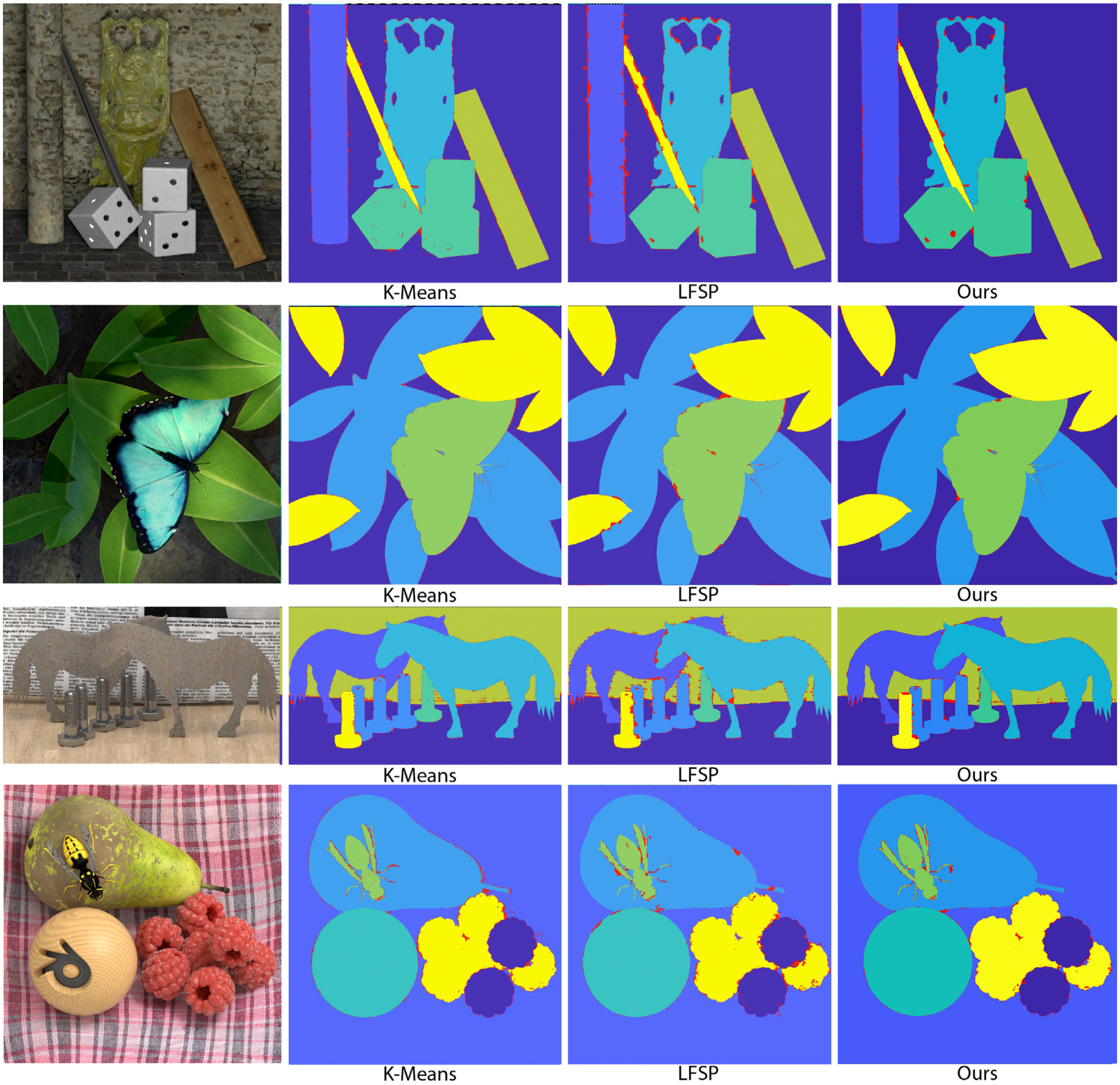
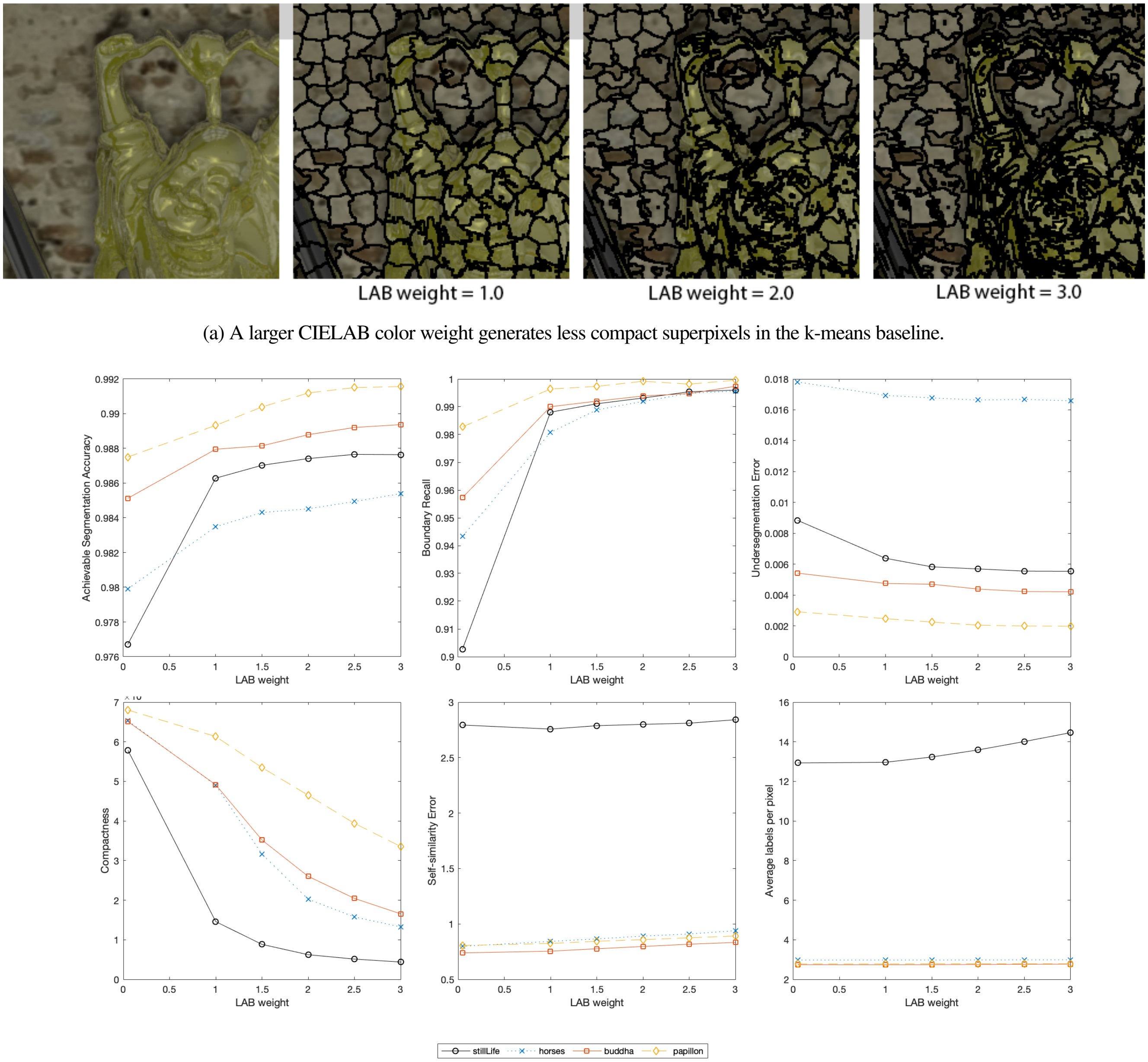


Figure 5: A visualization of the achievable segmentation accuracy in the central light field view. Red regions are superpixel sections which cross ground truth object segmentation boundaries. All other colors denote object labels.



(a) A larger CIELAB color weight generates less compact superpixels in the k-means baseline.

(b) Increasing the CIELAB color weight improves performance on traditional 2D superpixel metrics but decreases compactness, while slightly degrading performance on the light field specific metrics. Note that, as the *average labels per pixel* metric does not include non-central occluded pixels, this baseline performs strongly (as discussed in the main paper).

Figure 6: An evaluation of the effect of feature weight on clustering, as a demonstration of the trade-off between boundary following and spatial/depth regularity. Here, we show results of the k-means-based clustering baseline with Wang et al. depth on the HCI dataset. The superpixel size is fixed at 20.

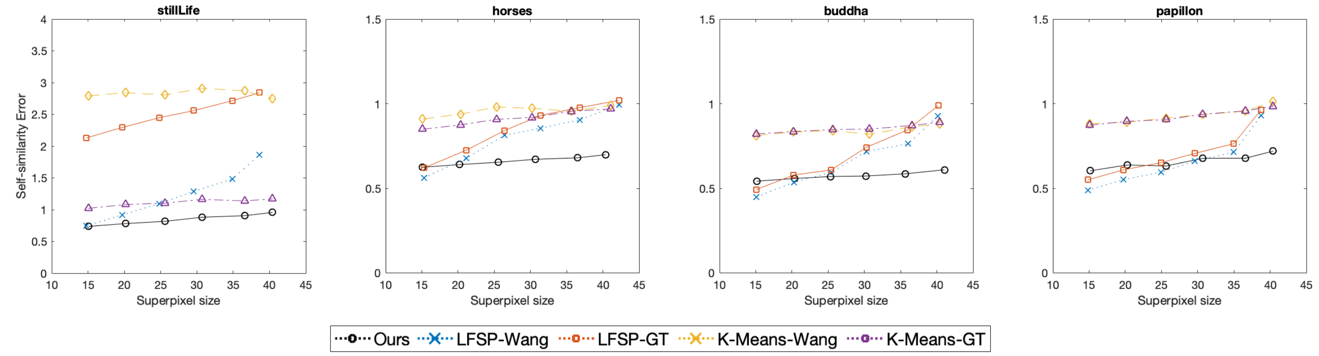


Figure 7: Self-similarity error measured over all light fields in the HCI dataset. This error provides a measure of the consistency of superpixel shape across views; smaller errors indicate greater consistency.

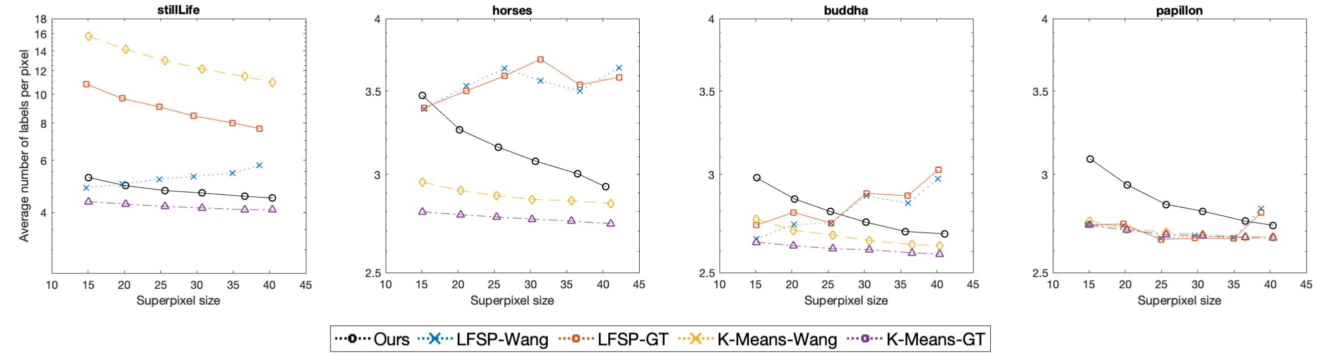


Figure 8: The average number of labels per pixel measured over all light fields in the HCI dataset. Smaller values indicate that a greater number of pixels have a consistent label across views.

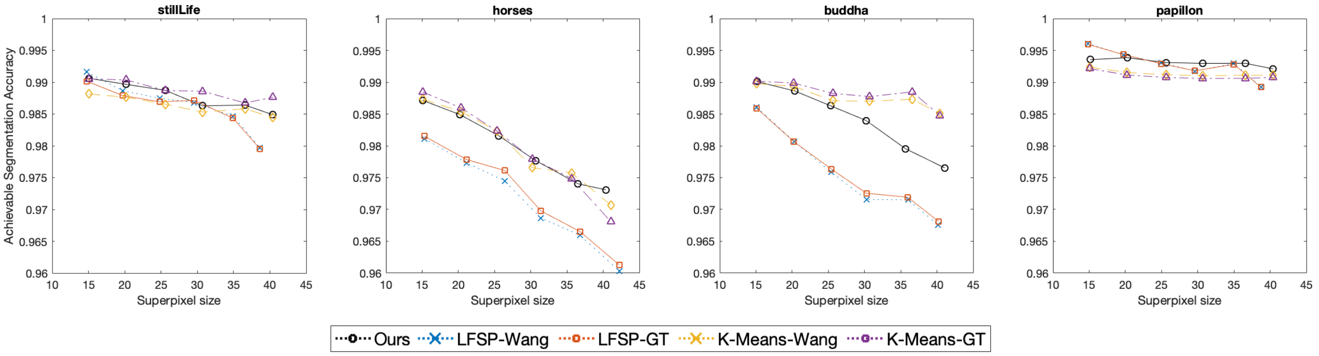


Figure 9: The achievable segmentation accuracy measured over all light field in the HCI dataset. The achievable segmentation accuracy describes how well the segments align with the ground truth labels. Hence, it provides a measure of the percentage of correctly labeled pixels. While ASA is not the same as object segmentation accuracy, it provides an upper bound on the accuracy of an object-level segmentation based on the current oversegmentation.

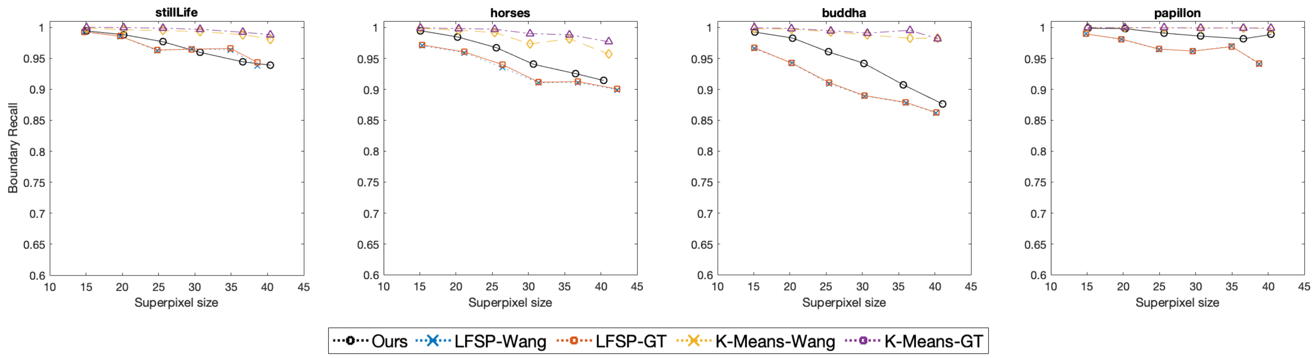


Figure 10: Boundary recall measured over all light fields in the HCI dataset. Boundary recall measures the fraction of ground truth edges which fall within a distance d of one or more super pixel boundary. Intuitively, the higher the boundary recall, the better the superpixels adhere to object boundaries.

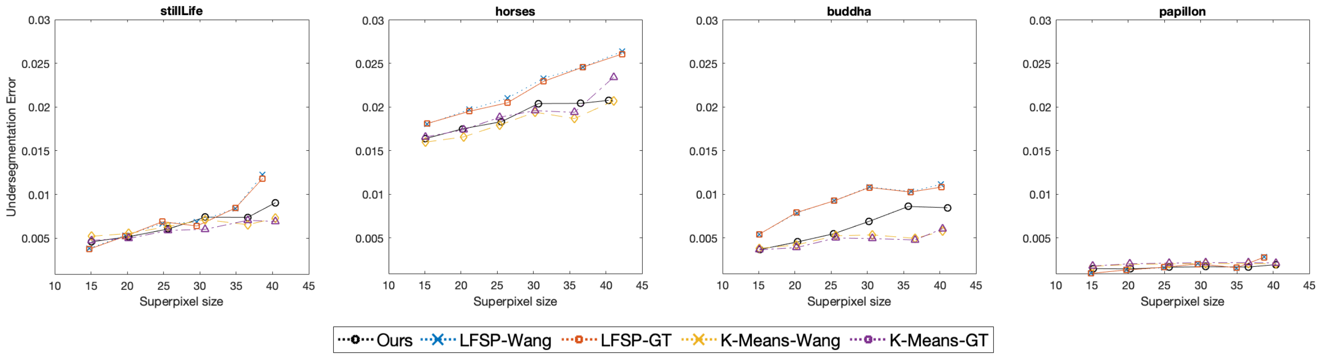


Figure 11: Undersegmentation Error measured over all light fields in the HCI dataset. Undersegmentation error measures the percentage of superpixels that extend over ground truth segment borders

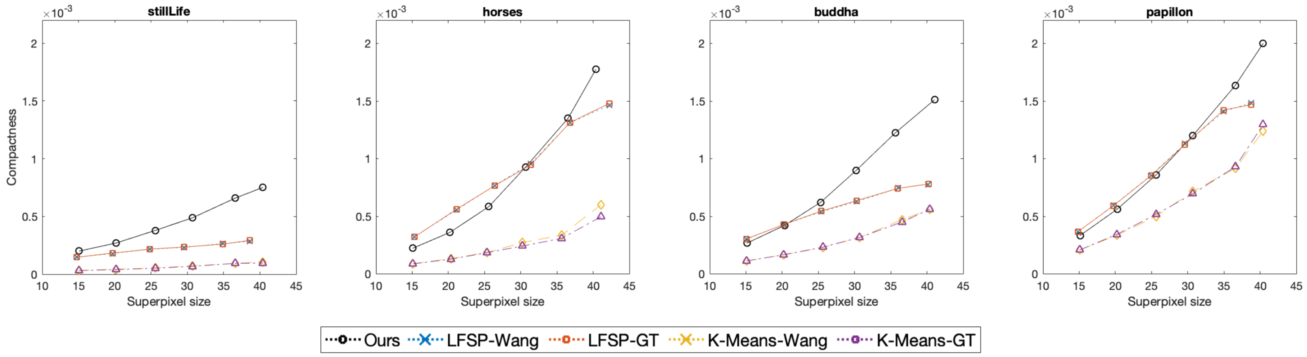


Figure 12: Compactness measured over all superpixels in the light fields of the HCI dataset. The compactness is related to the ratio of area to perimeter, and larger values signify smoother superpixel boundaries.