5 - Kinect Depth Based Video Segmentation

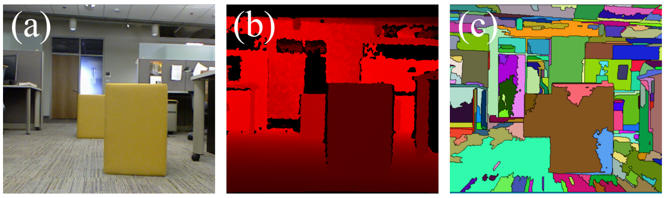


Fig. 1. **(a)** shows an image of two objects with similar appearance but at different depths. **(b)** gives the Kinect depth map. **(c)** shows segmentation problems in [3] due to lack of depth cues (the yellow objs. were combined).

I. Introduction

Dense video segmentation concerns many vision researchers since it provides a useful low-level cue for many problems. Traditionally, methods simply observe colors when combining voxels to make space-time regions. In video, optical flow can also play a role in discriminating regions. While the combination of these descriptors can achieve reasonably accurate segmentations, algorithms fail when multiple objects are of similar color undergoing the same motion. With the decreasing cost of consumer stereo cameras, it has become increasingly popular to have RGBD video, which synchronizes both RGB and depth information in every frame. We will develop a video segmentation algorithm that uses RGBD readings from a Microsoft XBOX Kinect. This will enable our algorithm to distinguish between two regions that have similar appearance and motion but lie across a depth discontinuity.

In this paper we compare the efficacy of different descriptors employing depth information with appearance and flow. For each descriptor, we evaluate different distance metrics to find which complement motion and intensity measures. We quantitatively compare the segmentations produced with these metrics on ground-truth labellings for depth varying objects. We also qualitatively analyze our results against segmentations done with no depth information. Overall we demonstrate the value of explicit depth information in video segmentation frameworks, where motion and appearance cues fail. Having shown the advantages of depth cues, we also discuss possibilities for achieving similar performance in the absence of a depth sensor.

II. Related Work

Most methods for dense video segmentation are based on photometric consistency [9], [1]. Apart from using color, Grundmann *et al.* [3] uses optical flow explicity to provide a hierarchy of spacetime superpixels. For capturing non-local image characteristics in a computationally efficient way, their initial segmentation is based on the graph-based approach of [2]. In our algorithm, we will augment [3] with depth information from the Kinect.

Other proposed techniques demonstrate the importance of occlusions in creating space-time superpixels [5], [8]. We combine the framework from [3] with the occlusion reasoning of Humayun *et al.* [4] in an attempt to understand how monocular occlusion cues in RGB videos can be used to replace the extra depth sensor.

III. Methodology

We incorporate depth information by extending the framework of Grundmann *et al.* [3]. This algorithm runs in two stages. In the first step it oversegments the space-time volume using Felzenszwalb *et al.* [2], where temporal edges are directed by optical flow. In the second step, supervoxels created in the first step are combined by observing the χ^2 distance between color and flow histograms. By adjusting the τ merging threshold [2] and the minimum allowed region size, they create a hierarchy of segmentations. Each stage in the hierarchy aims to merge regions from the stage preceding it. This algorithm works well when appearance and flow cues are enough for following region boundaries, but breaks down when two nearby objects move similarly or appear the same. In such situations the human visual system (HVS) heavily relies on stereo depth to separate out segments in the scene. Taking inspiration from the HVS, we augment the video segmentation framework by incorporating depth information from a Kinect sensor. We make use of its inbuilt RGB camera to avoid any need for calibration.

Depth is used in both the oversegmentation and the hierarchical stage of the algorithm. For the oversegmentation stage, the edge weight between pixels i and j are computed as

$$w(e_{ij}) = \alpha \cdot |I_i - I_j| \; + \; (1-\alpha) \cdot |D_i - D_j| \quad , \qquad (1)$$

where I and D are the RGB and the Depth image. We set $\alpha = 0.5$ for our experiments. Note that this edge function is only used for spatial edges, since we expect depth for objects to vary across time and using it as a temporal cue might confuse the algorithm. For the hierarchical stage, we change the edge weight between two regions k and l to

$$w(e_{kl}) = (1 - (1 - \langle I_k, I_l \rangle)\,(1 - \langle F_k, F_l \rangle)\,(1 - \langle D_k, D_l \rangle))^2 \quad , \qquad (2)$$

where $\langle I_k, I_l \rangle$ is the image intensity distance metric. This edge weight also caters for flow (denoted by $\langle F_k, F_l \rangle$) and depth differences. The following sections explain the different depth descriptors and associated distances for this region edge weight.

A. Depth Descriptors

Before different regions can be compared based on depth, we need an efficient way to describe these measurements. Although it is theoretically possible to use raw depth per pixel for comparing regions, this scheme is extremely expensive when evaluating millions of pixels present in the video volume. This reason coupled with the need to deal with noise, we compress the depth information into summarized descriptors. Since these descriptors are used to make decisions when to merge regions, they are updated whenever two regions merge.

1) Average Depth: Most objects which are merged due to confusing flow and appearance lie at widely different depths. In such cases it is usually sufficient to use the average depth values of all pixels in a region for making merge decisions. Note that average depth can be misleading for surfaces which are not parallel to the focal plane. This requires more revealing depth descriptors, one of which could be histograms.

2) Histograms: In one set of experiments we discretize depth to be stored in histogram. This representation allows more flexibility as it can (1) represent depths of surfaces not parallel to the focal plane; (2) deal with noisy measurements.

B. *Depth Distance Metrics*

We evaluated four different distance metrics for comparing the depth descriptors given in the preceding section. These metrics are applied to Equation 2 (as $\langle D_k, D_l \rangle$) to incorporate depth cues with flow and appearance. Note that the global distance metric given in Equation 2 sets $\langle D_k, D_l \rangle = 1$ whenever Kinect reports no depth values for either region k or region l. This heuristic forces the algorithm to rely only on flow and color information in such circumstances.

1) ℓ-norm Distance: Given the average depth of voxels of two regions, we can compute the ℓ_1 depth distance between these two. This rather simplistic approach works well wherever the depth disparity between neighboring regions is large. When we merge two regions in this approach, the new depth descriptor can simply be their average. We improve this descriptor update by using a size weighted average

$$D_{k \cup l} = D_k \left(\frac{n_k}{n_k + n_l} \right) + D_l \left(\frac{n_l}{n_k + n_l} \right) \quad , \qquad (3)$$

where n_k and n_l are the number of voxels in the respective regions. $D_{k \cup l}$ is the average depth of the new merged region.

2) χ^2 Distance: This distance metric can be used to compare normalized histograms by summing the chi-squared error over all bins in the histogram:

$$\langle D_k, D_l \rangle = \frac{1}{2} \sum_{b=1}^{B} \frac{(h_k(b) - h_l(b))^2}{h_k(b) + h_l(b)} \quad , \qquad (4)$$

where B is the number of bins and $h_k(b)$ is the normalized value of region k at bin b. This is known as a bin-to-bin comparison.

3) EMD-ℓ_1 Distance: This distance metric can be used to compare normalized histograms by:

$$\langle D_k, D_l \rangle = \min_{G=\{g_{i;j}:(i,j)\in\mathcal{J}_1\}} \sum_{\mathcal{J}_1} g_{i;j} \quad , \qquad (5)$$

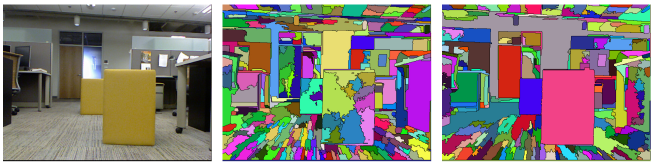
where G is a flow from D_k to D_l and $g_{i;j}$ denotes flow from bin (i) to (j). This is known as a cross-bin comparison as presented in Ling *et al.* [6].

IV. Evaluation

We test our algorithm using four different sequences, each focused on the specific merge between two objects that are of similar appearance, but are located at different depths. Our results are subjected to a qualitative analysis that seeks to compare the accuracy of the merges for different metrics at different hierarchical levels. Figure 3 shows the results for each test sequence and each metric. The left-most column is the RGB for the specific frame being compared. Each of the other columns represents the algorithm without depth, the algorithm with size-weighted average depth, the chi-squared histogram distance, and the earth mover's distance histogram distance, in that order. For testing purposes, each histogram metric uses histograms of 100 bins. A quantitative analysis was performed on the run-time and hierarchical step of merging for each different metric. The values are given in Table I for each sequence and metric.

	RGB	No Depth	W. Avg. Depth	χ^2	EMD-ℓ_1
RadsShirt	26.72	26.69	26.84	26.81	26.99
StaticDrawers	10.63	10.10	10.12	10.14	10.19
SlidingDrawers	44.92	44.65	44.76	44.83	45.35
TennisBalls	15.72	16.16	16.19	16.30	16.53

TABLE I
Run-time for different sequences.



(a) RGB (b) No Depth (c) With Depth

Fig. 2. Shows how the oversegmentation changes with inclusion of depth. The two yellow objects are broken to many supervoxels when using no depth information, whereas when using Equation 1, the objects are clustered correctly (blue and pink segments).

V. Discussion

During the oversegmentation stage, there is a distinct difference when depth is present. Figure 2 shows the oversegmentation for both cases. When depth is present, more merges are made between regions that have similar depth and color. This is most noticable in the yellow drawers, the door, and the wall. As evidenced in Figure 3, depth as a feature descriptor in the hierarchical stage can also provide more accurate segmentation of objects that are similar in color and/or optical flow.

The most important thing to note about the run-time values in Table I is that the difference in run-time between each of the four distance metrics is negligable for short clips. This occurs because the majority of the computational cost exists in the pre-processing and oversegmentation stages. Another important thing to note is that when depth is used in the algorithm, the run-time can actually decrease. This may seem counter-intuitive but can occur because the presence of depth influences more regions to merge in the oversegmentation step as shown in Figure 2. If fewer regions are fed to the hierarchical step, less merging needs to take place, decreasing the computational cost.

Table II represents how each algorithm performs in the hierarchical step for each sequence. In every sequence there are two object that have very similar appearance (purple shirts, yellow drawers, and green tennis balls). The hierarchical step at which the two objects merge is given in the table. An 'N/M' in the table represents when the two objects never merge at any level. As shown, the chi-square histogram distance out performs all approaches.

The difference between the chi-squared histogram distance and the earth mover's distance histogram distance metrics is clearly displayed in Figure 3. Because chi-squared only compares bins at each index of the histograms directly, it functions well for objects that occupy only a few depth bins, like planar surfaces that are perpendicular to the camera. The earth mover's distance out-performs the chi-squared algorithm when objects occupy multiple depth bins. An example of this is the floor in Figure 3 row 2 where it merges

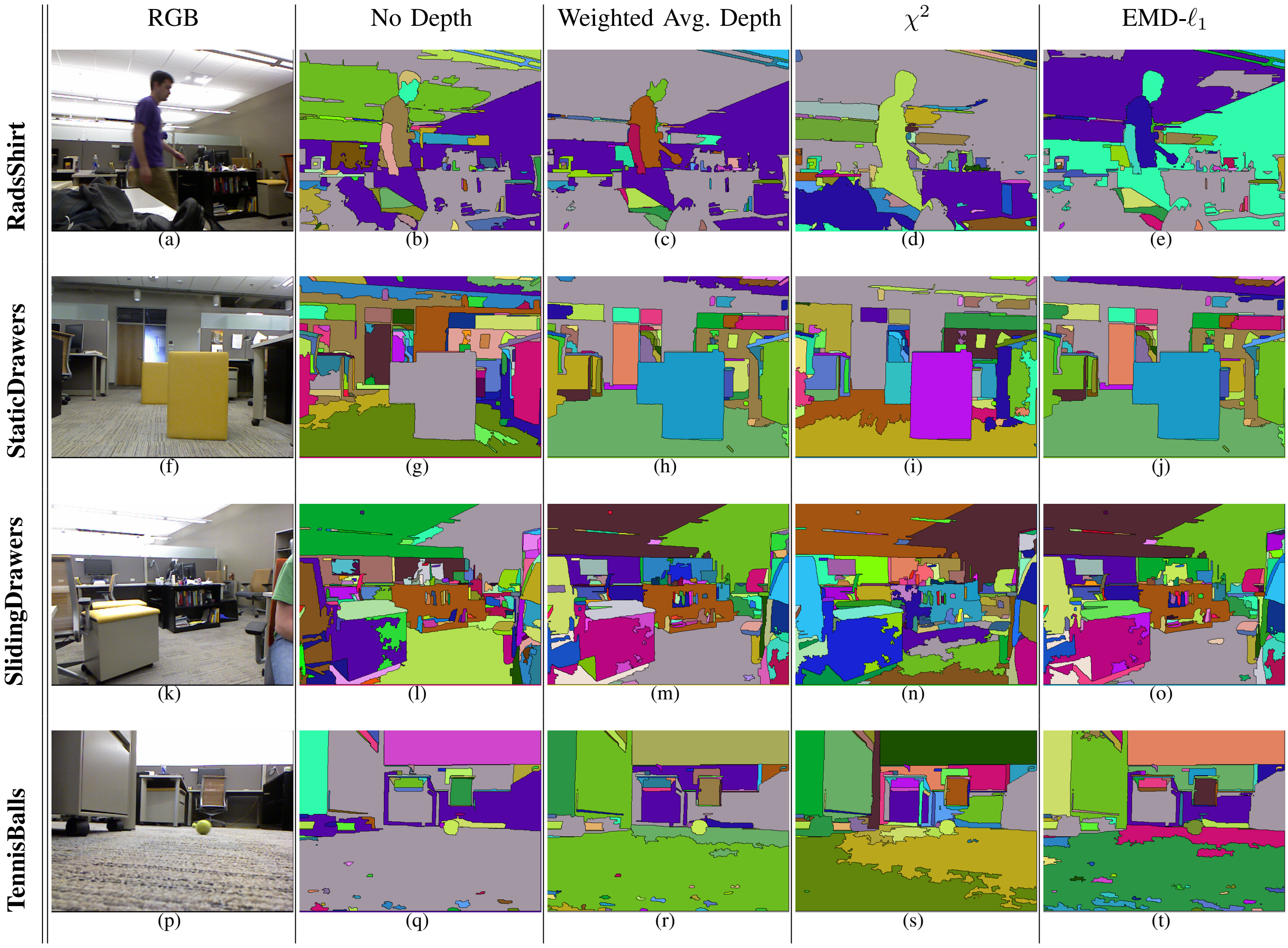


Fig. 3. Qualitative results. Each row shows segmentation results for a particular frame in a sequence.

	No Depth	W. Avg. Depth	χ^2	EMD-ℓ_1
RadsShirt	5	9	11	9
StaticDrawers	2	4	N/M	4
SlidingDrawers	3	3	13	3
TennisBalls	15	N/M	N/M	8

TABLE II
HIERARCHY LEVEL AT WHICH SIMILAR OBJECT MERGE OCCURS.

much more quickly together. The chi-squared algorithm fails here because the regions have non-overlapping populated histogram bins.

This work has confirmed the idea that depth from a Kinect sensor can be used as a feature descriptor to increase the accuracy of segmentation. The results reveal the fact that in terms of computational cost, the different distance metrics are very similar. The size-weighted average, chi-squared histogram distance, and the normalized earth mover's distance histogram distance are the most accurate and each should be used according to the nature of the scene that is to be segmented. Future work will use [7] for performing relative depth reasoning on monocular sequences. This will be performed by using occlusion cues in conjunction with region shape moments to determine ordinality of the regions in the oversegmentation, then using the ordinality as a feature descriptor in a second pass of the algorithm. The accuracy of this segmentation will compared the results of this paper in order to see how much depth information we can obtain from a monocular sequence.

REFERENCES

[1] F. Drucker and J. MacCormick. Fast superpixels for video analysis. In *WMVC*, pages 1 –8, dec. 2009.

[2] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59:167–181, 2004.

[3] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *IEEE CVPR*, pages 2141 –2148, june 2010.

[4] A. Humayun, O. Mac Aodha, and G. J. Brostow. Learning to find occlusion regions. In *IEEE CVPR*, pages 2161 –2168, june 2011.

[5] J. Lezama, K. Alahari, J. Sivic, and I. Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *IEEE CVPR*, pages 3369 –3376, june 2011.

[6] H. Ling and K. Okada. An efficient earth mover's distance algorithm for robust histogram comparison. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29:840–853, May 2007.

[7] K. Nakayama, S. Shimojo, and G. Silverman. Stereoscopic depth: its relation to image segmentation, grouping, and the recognition of occluded objects. *Perception*, 18(1):55–68, 1989.

[8] A. S. Ogale, C. Fermüller, and Y. Aloimonos. Motion segmentation using occlusions. *IEEE PAMI*, 27(6):988 –992, june 2005.

[9] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. Multiple hypothesis video segmentation from superpixel flows. In *ECCV*, volume 6315 of *Lecture Notes in Computer Science*, pages 268–281. Springer Berlin / Heidelberg, 2010.