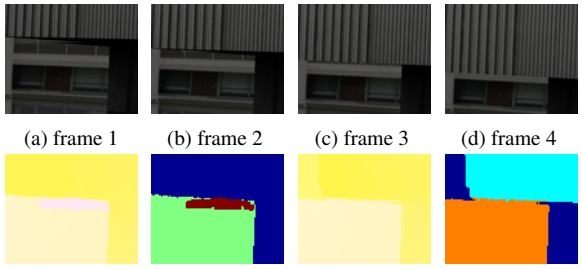
Table 1. Average end-point error (EPE) on the Middlebury *training* set. Using four frames and the new optimization improves accuracy.

	Avg.	Venus	Dimetrodon	Hydrangea	RubberWhale	Grove2	Grove3	Urban2	Urban3
Classic+NL (2 frames)	0.221	0.238	0.131	0.152	0.073	0.103	0.468	0.220	0.384
Layers++ (2 frames)	0.195	0.211	0.150	0.161	0.067	0.086	0.331	0.210	0.345
Layers++ (4 frames)	0.190	0.211	0.151	0.157	0.067	0.084	0.330	0.207	0.311
nLayers (4 frames)	0.183	0.191	0.126	0.175	0.062	0.080	0.336	0.175	0.316

Table 2. Average end-point error (EPE) and angular error (AAE) on the Middlebury optical flow benchmark *test* set. The discrete-continuous optimization (**nLayers**) obtains similar EPE and better AAE than the continuous-only inference method (**Layers++**).

		Rank	**Avg.**	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Teddy
EPE	**Layers++**	8.0	0.27	0.08	0.19	0.20	0.13	0.48	0.47	0.15	0.46
	nLayers	8.5	0.28	0.07	0.22	0.25	0.15	0.53	0.44	0.13	0.47
AAE	**Layers++**	9.2	2.56	3.11	2.43	2.43	2.13	2.35	3.81	2.74	1.45
	nLayers	5.7	2.38	2.80	2.71	2.61	2.30	2.30	2.62	2.29	1.38



(a) frame 1 (b) frame 2 (c) frame 3 (d) frame 4

(e) flow (2 to 3) (f) seg. (2) (g) flow (2 to 3) (h) seg. (2)

Figure 5. Occlusion reasoning using frames 2 and 3 (e-f) is hard (detail from Urban3); enforcing temporal coherence of the support functions using 4 frames significantly reduces the errors in both the flow field and the segmentation (g-h). The flow field is from frame 2 to frame 3 and the segmentation is for frame 2.

as **HGVS** in the comparison below. **HGVS** uses the output from a recent optical flow estimation method [34].

Implementation details and parameter settings. We start with the single-layered output from "Classic+NL" [26] and cluster the flow field into 10 layers. We then run the discrete method to estimate the scene structure and the flow fields to initialize the more precise continuous layered model. It takes **nLayers** about 10 hours in total to compute three forward and three backward flow fields from the four-frame 640×480 "Urban" sequence in MATLAB with a C++ mexed QPBO solver. It takes **Layers++** about 5 hours to compute one forward and one backward flow field from two frames. **HGVS** uses ten frames, or all the frames if a sequence has fewer than ten frames. **HGVS** has three different outputs for the same video. We show the segmentation results produced at 90 percent of highest hierarchy level, because it gives the best visual and numeric results.

4.1. Motion Estimation

We use the Middlebury optical flow benchmark to evaluate the motion estimation results. We manually set $\lambda_{\text{aff}} = 0.3, \lambda_{\text{b}} = 80$, and $\lambda_c = 10$ for the discrete model, use the provided values for the other parameters from [27], and fix them for all the motion estimation experiments. We set all the robust functions to be the generalized Charbonnier penalty function $\rho(x) = (x^2 + \epsilon^2)^a$ with $\epsilon = 0.001$ and $a = 0.45$ [26].

Results on the Middlebury *training* set are shown in Table 1. Changing from 2 to 4 frames improves results for the **Layers++** model supporting our hypothesis that longer sequences are important. More improvement comes from using a discrete model to obtain a good segmentation of the scene and then use the inferred structure for flow estimation (**nLayers**, 4 frames). Figure 5 shows a case where using 4 frames resolves ambiguity in the layer assignment and reduces errors in the estimated motion.

On the *test* set, **nLayers** obtains EPE similar to **Layers++** but better AAE, as shown in Table 2. At the time of writing (April 2012), **nLayers** is ranked first in AAE and fourth in EPE (see **Supplemental Material** for the screen shot). AAE measures the angle between the estimated motion vector and the ground truth and EPE is the Euclidean difference between the two. The results suggest that **nLayers** estimates motion directions more accurately.

Figure 6 shows the estimated segmentation and flow fields on some test sequences. Nearly all the major structures of "Urban" are correctly recovered, resulting in the best boundary EPE and AAE performance. The higher overall error results from the bottom left building. A major part of the building moves out of the image boundary and has no data term to estimate the motion. **nLayers** uses the affine model to interpolate the motion of the out-of-boundary pixels, but the building's motion violates the affine assumption.

4.2. Layer Segmentation

The Middlebury dataset does not have motion segmentation ground truth and so we use the MIT human annotated dataset [18] to evaluate segmentation performance. Segmentation accuracy is computed using the RandIndex measure [22] (larger is better). Because the MIT dataset is different in nature from the Middlebury dataset and has more rigidly moving, distant objects, we use a larger weight on the affine unary term as $\lambda_{\text{aff}} = 1$, and $\lambda_c = 3$ for the dis-