CS7495 - Portfolio

Ahmad Humayun
School of Interactive Computing
Georgia Institute of Technology, Atlanta, Georgia 30332–0280
Email: ahumayun@cc.gatech.edu

I. About me

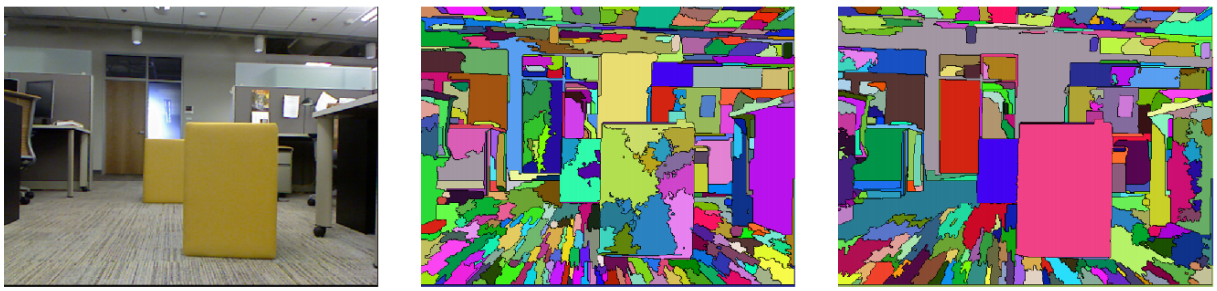
- I am a Computer Vision researcher working with Jim Rehg and Irfan Essa as a PhD student at the Computational Perception Lab., Georgia Tech.
- Homepage: http://www.cc.gatech.edu/ ahumayun/

As a vision researcher, what excites me, and hopefully many others, is that we know of at least one exceptional vision system: the human brain - which theoretically gives us a perfect algorithm to mimic. I work with video data, since we humans have developed our perception from a continuous stream of frames - not from disparate images. Over the coming years (maybe decades, who knows?) I would like to build machines which can learn to perceive video while being extensible to even comprehend single images.

Recently I worked with Radford Parker over an experiment to see if depth is an essential cue for video segmentation. I am sharing my experience on this project since I think it brought up interesting questions about mid-level cues for scene understanding in videos. I'll breifly explain the thoughts behind the project and give any insight we gained during its course. I have tailored this document to be an easy read for fellow researchers in my community.

II. Monocular Occlusion Based Video Segmentation

The idea for this project developed from the questions we posed to ourselves in a previous project. Let me first tell you what the latter was about. The basic idea behind it was driven by curiosity to see if the current state-of-the-art video segmentation can be improved with depth cues. Dense video segmentation is considered a useful first step for many vision applications. It creates spacetime superpixels which help avoid processing millions of individual voxels. Traditionally, video segmentation methods work by simply observing low-level pixel cues when combining voxels to make space-time regions. This is typically done through appearance and optical flow cues. One thing amiss from these cues is the notion of depth in a scene. Depth can play a critical role in delineating object boundaries, especially where motion and appearance cues fail. Our hypothesis was inspired by human vision where depth disparities are important cues for segmentation of a (video) scene. By augmenting Grundmann *et al.* [2]



(a) input RGB (b) oversegmentation without depth cues (c) oversegmentation with depth cues

Fig. 1. Sample results from our initial project

using the Kinect as depth sensor, we got reasonable success in the second project. Figure 1 shows an example result.

It is noticeable that the number of false positives was reduced in the oversegmentation. I think this is one of the key ideas that the HVS uses to reduce the number of plausible explanations for a scene. Following this project, we asked ourselves if similar results are possible without using an explicit depth sensor. We concluded that there are two ways to include depth information in a video segmentation pipeline: (1) with the inclusion of an RGBD sensor or (2) by inferring ordinal depths through occlusion boundaries [3], [7]. Even though the former solution simplifies the problem, it is less elegant since it does not allow processing of monocular RGB videos. In this final project we improved the state-of-the-art video segmentation algorithm by detecting occlusion regions to infer depth boundaries. We evaluated our results by comparing the oversegmentations when using (1) no depth, (2) an RGBD sensor, and (3) occlusions to infer object boundaries.

A. *Related Work*

Most methods for video segmentation are based on photometric consistency [8]. Apart from using color, Grundmann *et al.* [2] uses optical flow explicity to provide a hierarchy of spacetime superpixels using Felzenszwalb and Huttenlocher [1] oversegmentation. Their algorithm trades the use of high-level cues like occlusions for computational efficiency. On the other hand, some techniques demonstrate the importance of occlusions in creating space-time superpixels [6].

A variety of methods exist for computing occlusions in frame pairs, including those based on graph-cuts [5] Recently, our paper (Humayun *et al.* [4]) introduced a learning framework to infer occlusions on a per pixel basis. In this project we used latter, in combination with [2], to improve video segmentation in monocular video.

B. *Approach*

Our algorithm had 3 steps: (1) compute forward and backward flow for a consecutive pair of frames from the input video; (2) compute occlusion for all pixels in the first frame; (3) use flow and occlusions to make space-time superpixels. Figure 2 gives an overview of our approach.

When a foreground object in a scene moves, some portion of the background gets occluded. This information is relevant for segmentation because two regions sharing an occlusion boundary need not be merged because they belong to different objects. Humayun *et al.* [4] uses features based on appearance, texture, and flow to train a random forest for finding pixels that are occluded. Since the label here is binary (”occluded”, ”not occluded”), the forest performs regression to produce a probability of occlusion $\mathcal{O}_i$ for pixel i.

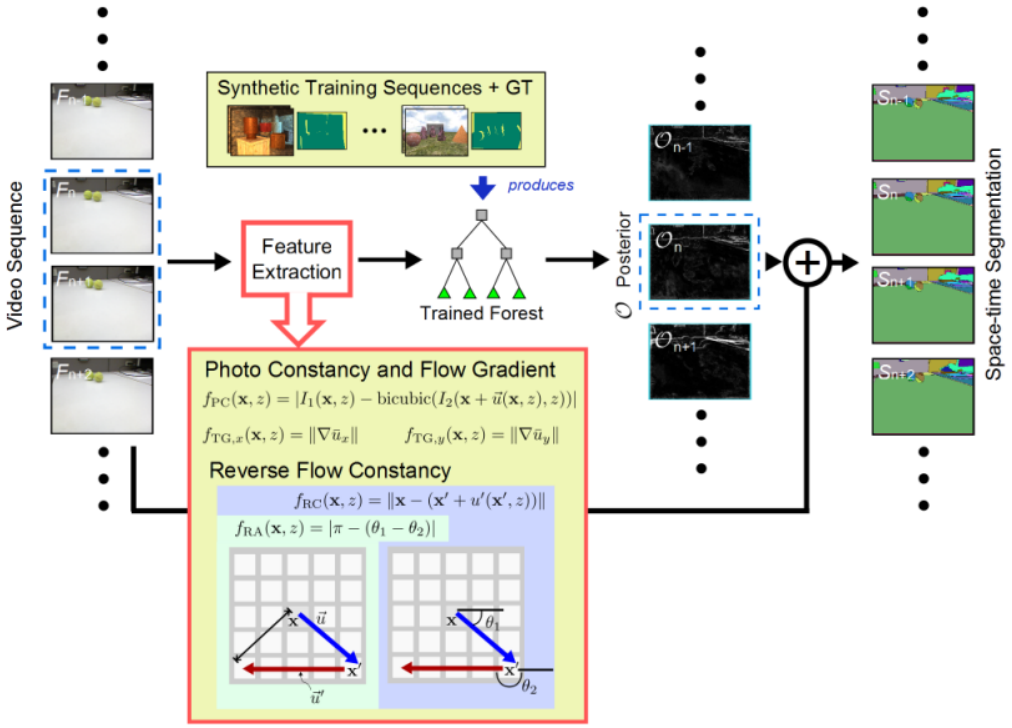


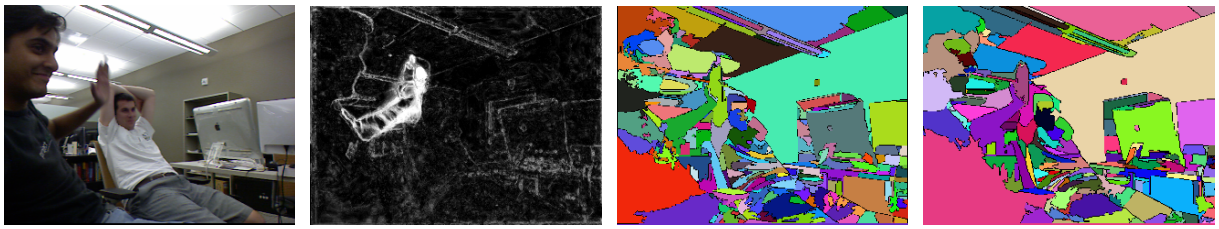
Fig. 2. Pipeline of our system.

Part of our research was to decide which features used by [4] can be discarded in lieu of efficiency without sacrificing a significant amount of precision. We experimented varying combination of features proposed in [4], by taking hints from the feature importance returned by the forest. Although [4] shows improvements in classifying occlusion with the use of multiple flow algorithms, we constrain ourselves to a single flow algorithm [9] in order to reduce computation time. The features we used were based on temporal gradients, photo-constancy and reverse flow direction (as given in the figure above).

During the oversegmentation stage, Grundmann *et al.* [2] uses the ℓ_2 color distance to create edge weights between pixels both spatially and temporally. Some of the edge weights that lie across object boundaries are incorrectly assigned a low score due to the objects being of similar color. At these edges, the graph cut algorithm of Felzenszwalb and Huttenlocher will produce false negatives causing the objects to be merged. Because this edge does not appear in the oversegmentation stage, the error will be propagated to every level of the hierarchical segmentation and cannot be recovered. In order to ensure that this edge is created, we used the per-pixel occlusion probabilities of Humayun *et al.* [4] as a distance cue in the oversegmentation. Essentially, pairs of pixels with a large occlusion probability difference are assigned a larger weight and will not be combined while we segment the graph. The opposite is true for pairs of pixels with a small occlusion probability difference; they are assigned a smaller weight and should be combined together. This formulation is a result of the intuition that if occlusions are found between two seperate regions, they must belong to different objects and should not be merged. This information then gets propagated in order to preserve the edge in the hierarchy.

C. Results

The feature selection process involved determining the features that yield the most accurate occlusion classification of the pixels. We evaluated the classifier in each case with Maya-



(a) input RGB (b) occlusions (c) oversegmentation without occlusion information (d) oversegmentation with occlusion information

Fig. 3. Sample results from our initial project

generated sequences. By quantitatively evaluating the F1 scores from precision-recall curves for each combination of features across all test sequences, we determined the most accurate combination of features.

The performance of the occlusion-based oversegmentation was compared to the original oversegmentation and to the oversegmentation with depth cues from an RGBD sensor. The algorithm was tested using real sequences where false negatives arise during the oversegmentation stage. Figure 3 shows one comparative result from the project. Notice how the area around the raised hand erroneously gets combined with segments on different surfaces when using no occlusion information (Figure 3(c))- which is fixed by our method (Figure 3(d)).

Video results comparing the segmentation of two tennis balls with and without using occlusion information:

- Without occlusion information.
- With occlusion information.

D. Conclusion

Although our approach at times helps separate different objects, the process is detrimental whenever there are no occlusion cues between two objects (for instance the chair's arm in the result above). Overall, occlusions inferred from an RGB sensor can help dilineate object boundaries to some extent. We also observed that our resulting monocular oversegmentation implementation can often achieve better results than using a depth sensor when evaluated for how many hierarchies a true positive survives. A possible explanation for the monocular occlusion oversegmentation outperforming the RGBD oversegmentation is that the depth map is often noisy and contains incomplete areas.

Overall, we were pleased to see that monocular vision can provide similar cues to that of a depth sensor when it comes to video segmentation.

As said before, our hypothesis was inspired by human visual system, where depth plays an important role in reducing the number of plausible segmentations. What excited me about this project was that we were in position to test if a computer vision system could reason about space-time segmentations based simply on monocular cues. Even though humans employ stereo vision to get such cues, monocular video with ego-motion or scene motion can also reveal depth information - which we successfully used in this project. Just seeing a working system, loosely mimicing cues that the human vision possibly uses made this an exciting project.

REFERENCES

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

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

[3] X. He and A. Yuille. Occlusion boundary detection using pseudo-depth. In *ECCV*, volume 6314 of *LNCS*, pages 539–552. 2010.

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

[5] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *IEEE ICCV*, pages 508 –515 vol.2, 2001.

[6] 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, 2011.

[7] A. Stein and M. Hebert. Occlusion boundaries from motion: Low-level detection andmid-level reasoning. *IJCV*, 82:325–357, 2009.

[8] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. Multiple hypothesis video segmentation from superpixel flows. In *ECCV*, volume 6315 of *LNCS*, pages 268–281. 2010.

[9] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic Huber-L1 optical flow. In *BMVC*, 2009.