Geometric Context from Videos

S. Hussain Raza Matthias Grundmann Irfan Essa
Georgia Institute of Technology, Atlanta, GA, USA
http://www.cc.gatech.edu/cpl/projects/videogeometriccontext

Abstract

We present a novel algorithm for estimating the broad 3D geometric structure of outdoor video *scenes. Leveraging* spatio-temporal *video segmentation, we decompose a dynamic scene captured by a video into geometric classes, based on predictions made by region-classifiers that are trained on appearance and motion features. By examining the homogeneity of the prediction, we combine predictions across* multiple *segmentation hierarchy* levels *alleviating the need to determine the granularity a priori. We built a novel, extensive dataset on geometric context of video to evaluate our method, consisting of over 100* ground-truth annotated *outdoor videos with over 20,000 frames. To further scale beyond this dataset, we propose a semi-supervised learning framework to expand the pool of labeled data with high confidence predictions obtained from unlabeled data. Our system produces an accurate prediction of geometric context of video achieving 96% accuracy across main geometric classes.*

1. Introduction

Holistic scene understanding requires an understanding of the broad 3D structure of the scene with all objects present. One important step towards this goal is to partition a scene into regions and label them relative to each other and within the scene geometry. Geometric classes can define the basic 3D structure of a scene with respect to the camera, and suggest cues to identify horizontal surfaces and vertical objects in the scene. Hoeim *et al.* [12] showed that such geometric context can be used to obtain a probabilistic representation of the scene layout based on geometric classes, which in turn can be used to improve object detection. Torralba *et al.* [18] showed that global context plays an important role in object detection. Recently, Divala *et al.* [5] showed that incorporating geometric context, not only improves object detection but also makes misclassifications more reasonable.

In this paper, we propose a novel method to provide a high level description of a *video* scene by assigning geometric classes to spatio-temporal regions as shown in Figure 1.

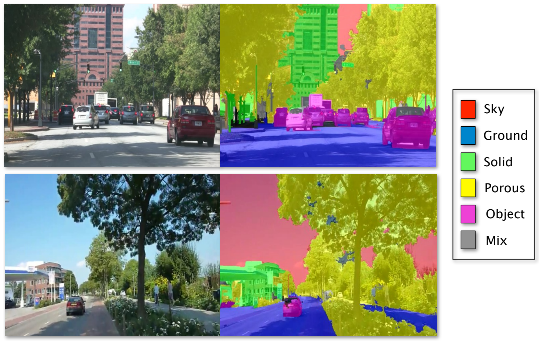


Figure 1: Video frames of an urban scene (left) and the predicted geometric context of our system (right). We achieve high accuracy leveraging motion and appearance features while achieving temporal consistency by relying on spatio-temporal regions across various granularities. Please watch the accompanying video.

Building upon a hierarchical video-segmentation to achieve temporal consistency, we compute a wide variety of appearance, location, and motion features which are used to train classifiers to predict geometric context in video. A significant challenge for developing scene understanding system for videos is a need for an annotated *video* dataset available for training and evaluation. To this end, we have collected and annotated a video dataset with pixel level ground truth labels for over 20,000 frames across 100 videos covering a wide variety of scene examples.

The primary contributions of this paper are:

- A scene description for video via geometric classes (96% accuracy across main geometric classes).
- Exploiting motion and temporal causality/redundancy present in video by using motion features and aggregating predictions across spatio-temporal regions.
- A semi-supervised bootstrap learning framework for expanding the pool of labeled data with highly confident predictions obtained on unlabeled data.
- A novel dataset of 100 annotated videos (∼20,000 frames) with pixel level labels, which will be made available.
- A thorough evaluation of our system by examining importance of features, benefit of temporal redundancy and independence of segmentation granularity.

2. Related Work

Image based scene understanding methods[13, 9] can be directly applied to individual video frames to generate a description of the scene. However, these methods do not exploit temporal information across neighboring frames. Further, lacking temporal consistency, they can result in temporally inconsistent labels across frames, which can impact performance, as scene labels suddenly change between frames. In addition, frame-based methods do not exploit temporal redundancy to process videos efficiently as processing each segment in video independently results in a longer processing time.

Temporal information can be incorporated using structure from motion (SfM) [16, 1], which requires substantial computation and might not generalize well to dynamic objects. SfM makes explicit assumptions about the scene, *e.g.* , mostly static with limited foreground motion, and requires basic camera priors. In addition, SfM reconstruction can fail due to lack of parallax, *e.g.* , walking forward, and rotation around camera center. We do not make any assumption about the scene content, amount of foreground motion, or the camera used. Another approach to achieve temporal consistency across frames is to use optical flow between consecutive frames to estimate the neighborhood of each pixel and then combine past predictions to make a final prediction [14]. This requires labeling every pixel in every frame in the video independently, which doesn't leverage the causality in video.

Our video scene understanding approach takes advantage of spatio-temporal information by employing hierarchical video segmentation[10], which segments a video into spatio-temporal regions. Further, we leverage causality in videos to efficiently label videos, achieving favorable complexity which is linear in the number of unique *spatio-temporal* segments in videos. Consequently, in contrast to image based or independent frame labeling, our system is not directly affected by the total number of frames. Recently, Tighe *et al.* [17] applied their image label transfer to the video domain leveraging[10], by applying a max heuristic across frames. In contrast, our approach performs geometric labeling by leveraging multiple hierarchy level while probabilistic aggregating labels over a temporal window.

A significant hurdle in video scene understanding is the availability of a ground truth annotated dataset for training. While several datasets exist for predicting geometric context in the *image* domain [13, 9], datasets for videos [2, 6, 19] are currently limited in their scope. (see section 3).

Our video scene analysis method builds upon Hoeim *et al.* 's [13] image based approach, extending the image based approach to video. Our approach differs, in that it is taking advantage of spatio-temporal context, extends feature set being more suitable for video, leverages temporal redundancy while achieving temporal consistency and broadens the pool of available data by semi-supervised learning.

3. Dataset and Geometric Classes

Existing Datasets: In our supervised learning setting, we require an annotated dataset supplying ground truth labels for training and evaluation. While several datasets for geometric scene understanding exists on still images [13, 9], our video-based scene analysis method demands an annotated video dataset. However, existing datasets for video scene understanding only provide limited ground truth data. The CamVid dataset [2] provides pixel-level labels for 701 non-consecutive frames (about every 30th frame, sampled at 1fps). The NYUScenes [6], and MPI-VehicleScenes [19] dataset consists of 74 and 156 annotated frames, respectively. Therefore, these datasets are not ideally suited for comprehensive studies. To overcome this limitation, we provide a novel, pixel-level annotated dataset for geometric scene analysis of video, consisting of over 20,000 frames across 100 videos.

A video dataset for geometric scene understanding: Our dataset consists of 160 outdoor videos, with annotations available for a subset of 100 videos. Some videos are collected from YouTube and others are recorded by us while walking or driving in an urban area. Video lengths range from 60 to 400 frames and resolution varies from 320×480 to 600×800, with varying aspect ratios. We partitioned the datasets into three sets: 63 videos are used for training and cross-validation (13,000 frames), 40 videos for independent testing via external-validation (7,000 frames), and 60 videos are kept unlabeled (14,000 frames) and are later used for semi-supervised learning (Section 5.1). Videos in the cross and external-validation sets are completely annotated with ground truth labels (every frame and pixel).

Videos in our dataset contain entities such as sky, ground, buildings, trees, and objects (cars, trains, humans). While many different partitions can be imagined, we follow [13, 11] and partition the video content into three main geometric classes: “Sky”, “support”, and “vertical”. To provide a more detailed description of the scene, we further divide the vertical class into three subclasses: “Solid”, “porous”, and “object”. The solid vertical sub-class includes *solid, static* objects resting on the ground, such as buildings, boards, bridges, and rocks etc. The porous vertical sub-class includes *non-solid, static* objects such as trees and foliage. Finally, *movable* objects, like humans, cars, boats, and trains are included in the object class. Notice, that in contrast to [13, 11] we do not account for the orientation of the vertical classes as their identity is likely to change due to camera motion in video. Table 1 gives an overview of the distribution of the classes in the cross-validation dataset, by showing the pixel area of each of the geometric classes.

Sky	32.5%
Ground	26.4%
Vertical	40.6%
mix	0.5%

(a) Main Classes

Solid	19.7%
Porous	15.6%
Object	3.7%

(b) Sub-vertical Classes

Table 1: Average area in pixels of each geometric class in the cross-validation dataset.

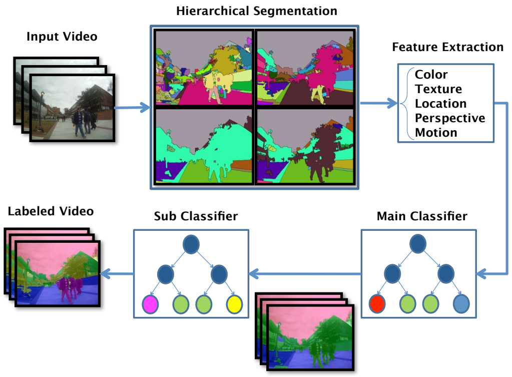


Figure 2: Overview of our approach. First, input videos are segmented into a hierarchy of spatio-temporal regions using [10]. Then, features are extracted for each segment to train a main and sub-classifier to predict geometric context in videos.

4. Geometric Context From Videos

Our algorithm for determining geometric context from video consists of 3 main steps (Figure 2). First, we apply hierarchical video segmentation, obtaining spatio-temporal regions at different hierarchy levels. We rely on video segmentation to achieve (a) temporal coherence without having to explicitly enforce it in our framework and (b) by labeling regions as opposed to individual pixels we greatly reduce computational complexity. Second, we extract several features from each segment. Third, we train a classifier to discriminate segments into *sky*, *ground*, and *vertical* classes. Additionally, a sub-classifier is trained to discriminate the vertical class further into *solid*, *porous*, and *object*. In particular, we employ a boosted decision tree classifier with a logistic version of Adaboost [3]. We will describe each of the above steps in more detail below.

4.1. Video Segmentation

Video segmentation aims to group similar pixels into spatio-temporal regions that are coherent in both appearance and motion. We use the hierarchical graph-based video segmentation algorithm proposed by Grundmann *et al.* [10, 20], which is automatic and achieves long-term coherence. For completeness, we give a brief overview of their algorithm. Their spatio-temporal hierarchical video segmentation builds upon the graph-based image segmentation of Felzenszwalb *et al.* [8] by constructing a graph over the 3-d space-time neighbors of a voxel. This approach generates an *over*-segmented video volume, which is further segmented into a hierarchy of super-regions of varying granularity. After computing region descriptors based on appearance and motion, a graph is constructed where each region from the over-segmentation forms a node and is connected to its incident regions by an edge with a weight equal to the χ^2-difference of their local descriptors. This so-called region graph is used to group the over-segmented regions into super-regions by applying [8] to the graph. Successive application of this algorithm yields a segmentation hierarchy of the video as shown in Figure 3 for one of our sample videos.

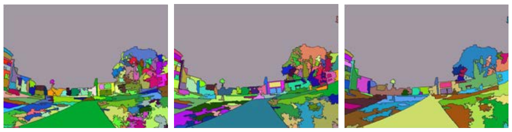


Figure 3: Video segmentation results by applying [10]. From left to right: Hierarchy levels in increasing order; region area increases as segments from lower hierarchy levels are grouped together.

4.2. Video Annotation

To obtain the ground truth for training and evaluation, we manually annotate over 100 videos. To greatly speed up the labeling process, we assign labels to individual spatio-temporal regions as opposed to pixels. In particular, we leverage the over-segmentation (Section 4.1) to assign the appropriate label to each supervoxel. Though errors in the over-segmentation are limited due to the fine granularity, we need to address potential *under-segmentation* errors, *i.e.* a supervoxel contains more than one class. This is particularly of concern for the vertical class which contains a wide variety of potentially overlapping surfaces, *e.g.* , buildings and trees, or several moving objects as cars, boats, trains, *etc*. To address this problem, we introduce a new label "mix" to label a super-voxel, which is a mixture of two or more classes or if its identity is changing over time across geometric classes. Figure 4 shows the labels and their hierarchical relationship.

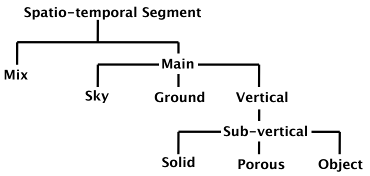


Figure 4: Annotation hierarchy of spatio-temporal segments: Segments are either labeled as either as a mixture of classes (mix) or assigned a main geometric class label. The vertical geometric class is further discriminated into solid, porous, and object.

To obtain a ground truth labeling for every level of the segmentation hierarchy, we leverage the ground truth labels of the over-segmented super-voxels, by pooling their

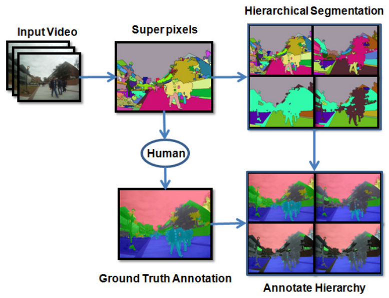


Figure 5: Video annotation for obtaining ground truth: Over-segmented super-voxels are annotated manually. Supervoxel labels are then combined to generate ground truth for each level of segmentation hierarchy (see Section 4.2).

Sky	2.5%
Ground	15.9%
Vertical	81.2%
Mix	0.4%

(a) Main Classes

Solid	47.5%
Porous	26.1%
Object	7.7%

(b) Sub-vertical Classes

Table 2: Percentage of segments annotated for each geometric class ($\sim 2.5M$ in total at over segmented base hierarchy level).

labels across a super-region via majority voting (a super-region is composed of several super-voxels). Specifically, if more than 95% of a super-region's area is assigned the same ground truth label L (based on the over-segmented super-voxels it is comprised of), the super-region is assigned label L, otherwise it is labeled as "mix", as shown in Figure 5. We manually annotated over 20,000 frames at the over-segmentation level and then combined their labels via the above approach across the hierarchy to generate labels at higher levels. Table 2 gives an overview of the percentage of segments annotated for each geometric class.

4.3. Features

We estimate the class-dependent probability of each geometric label for a segment in a frame using a wide variety of features. By segments, we refer to 2D per frame regions of the 3D spatio-temporal voxels. Specifically, we compute appearance (color, texture, location, perspective) and motion features *across* each segment in a frame. For computing appearance features, we follow Hoeim *et al.* [13] and apply the publicly available code on a per-frame basis. For details please refer to [13].

In videos, an additional feature not found in images is motion across frames. For motion features, we compute a histogram of dense optical flow (using OpenCV's implementation of Farneback's algorithm [7]) as well as the mean motion of a segment. To capture the motion and changes in velocity and acceleration of objects across time, we compute flow histograms and mean flow for each frame I_j w.r.t. to 3 previous frames: $I_{j-1}, I_{j-3}, I_{j-5}$. In particular, a segment S_k we compute a 16-bin histogram of oriented flow vectors weighted by their corresponding magnitude. Histograms are normalized by the region area of the segment in current frame. In addition, we compute histograms for spatial flow differentials in x and y, *i.e.* for the dense optical flow field $O = [O_x, O_y]$, we compute $[\partial_x O_x, \partial_x O_y]$ and $[\partial_y O_x, \partial_y O_y]$. To account for different scales, the flow differentials are computed for different kernel size of the Sobel filter (3, 5 and 7). As with the flow histograms, the spatial flow differentials are computed w.r.t. to 3 previous frames: $I_{j-1}, I_{j-3}, I_{j-5}$. This is similar to the approach of [4], which has been shown to to be helpful to the task of object detection in videos. Table 3 lists all of our motion based features used for estimating geometric context of video.

Motion based Features	
Dimensions	**Description**
16×3	Histogram of dense optical flow O of reference frame I_j w.r.t. $I_{j-1}, I_{j-3}, I_{j-5}$.
$16 \times 2 \times 3 \times 3$	Histogram of differential of dense optical flow O in x and y, *i.e.* $[\partial_x O_x, \partial_x O_y]$ and $[\partial_y O_x, \partial_y O_y]$, across 3 kernel sizes of differential (3, 5, and 7) for reference frame I_j w.r.t. $I_{j-1}, I_{j-3}, I_{j-5}$.
2×3	Mean flow of a segment minus min. mean flow across all segments of current frame.
2×3	Mean location change in x and y for reference frame I_j w.r.t. $I_{j-1}, I_{j-3}, I_{j-5}$.
$2 \times 2 \times 3$	10th and 90th percentile of location change in x and y for frame I_j w.r.t. I_{j-3}, I_{j-5}.
1×3	Magnitude of location change of mean, 10th and 90th percentile.

Table 3: List of flow and motion based features computed per frame I_j and per segment S_k. See text for details. Appearance features are adopted using the approach of Hoeim *et al.* [13].

4.4. Multiple Segmentations

As the appropriate granularity of the segmentation is not known a priori, we make use of multiple segmentations across several hierarchy levels, utilizing the increased spatial support of the segments at higher levels to compute features. In particular, we combine the individually predicted labels based on homogeneity of the segments. Homogeneity is defined in our case as the probability of the segment *not* being a mixture of several classes, *i.e. not* having the label mix (for details, see Section 4.5). We generate multiple segmentations of the scene at various granularity levels ranging from 10% to 50% of the hierarchy height using [10] in increments of 10% (5 hierarchy levels in total).

4.5. Classification

We evaluate our method using boosted decision trees based on a logistic regression version of Adaboost [3] that outputs the class probability for each segment in a frame and perform 5-fold cross validation. We train two multi-class classifiers to predict the geometric labels, first to discriminate

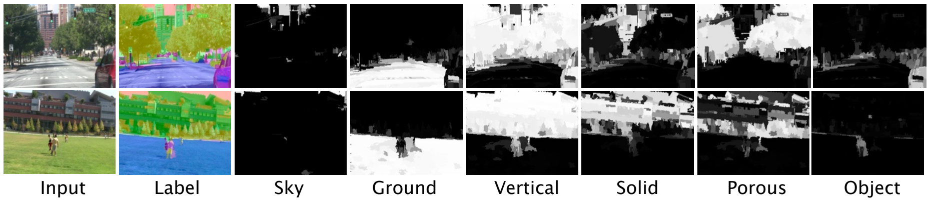


Figure 6: Input video image, predicted labels and confidence for each geometric class. Notice, that trees are correctly assigned high probability for porous class, walls for solid class, and humans and cars have high confidence for the object class.

among the main classes (sky, ground, and vertical), and second for further sub-classification of vertical class (buildings, porous, and objects). In addition to the two multi-class classifiers, we independently train a homogeneity classifier that estimates the probability of the segment being a single label segment or part of the class "mix". We refer to this probability as the *homogeneity* of a segment. This will enables us to take advantage of multiple segmentations, by combining the label confidence of segments based on their homogeneity. We combine the predictions of all three classifiers probabilistically to estimate the final label as described below.

Training: We extract the features described in Section 4.3 from each segment of a training video. As the segments vary across time, we opt to extract features for each frame for the same segment to provide discriminating information over time (*e.g.* appearance, motion, and pose of objects) as opposed to sampling features from unique spatio-temporal regions only. In addition, features are extracted independently for different hierarchical segmentation levels to provide instances with more spatial support. We extract features from 5 segmentation hierarchical levels ranging from 10% to 50% of the hierarchy height. Segments with a single ground truth label are used to train main and sub-vertical classifiers. We train the homogeneity classifier by providing examples of a single label and "mix" label segments as positive and negative instances.

Prediction: To predict the labels for a test video, features are extracted from each segment. A spatio-temporal region is labeled on a per-frame basis with the final classification being obtained by averaging the predicted class-posteriors across frames. We label main and sub-vertical geometrical classes independently, *i.e.* we compute the sub-vertical labels for all the segments in a frame but only apply it to segments labeled as vertical by main classifier.

When using multiple segmentations across different hierarchies, a super-pixel is part of different segments at each level of segmentation hierarchy. To determine the label y_i of *super-pixel* i, class-posteriors from all segments in the hierarchy s_j, containing the super-pixel are combined using a weighted average based on their homogeneity likelihoods $P(s_j|\mathbf{x}_j)$ [13, 11], where $\mathbf{x}_j$ is the corresponding feature vector. The likelihood of a segment label is then given as:

$$P(y_i = k|\mathbf{x}_i) = \sum_{j}^{n_s} P(y_j = k|\mathbf{x}_j, s_j)P(s_j|\mathbf{x}_j),$$

where, k denotes the possible geometric labels and n_s are number of hierarchical segmentations. This technique yields a final classification of super-pixels at the over-segmentation level by combining the individual predictions across hierarchy levels. These weighted posterior probabilities of super-pixels, for main and sub-vertical class, are then averaged across frames in a temporal window to give final predictions for each super-voxel.

5. Results

We report the accuracy of our method using the number of pixels correctly labeled on the testing videos, *i.e.* a 90% class accuracy indicates that 90% of the pixels of that class were labeled correctly. In our experiments, leveraging multiple hierarchy levels and temporal redundancy, we achieve an overall classification accuracy of *96.0%* for main and *77.4%* for the sub-vertical classes. After classification, each super-pixel is assigned the probability for each geometric class, as shown in Figure 6. Qualitative results are shown in Figure 7; we encourage the reader to watch the supplementary video.

	Sky	Ground	Vertical
Sky	99.4	0.0	0.6
Ground	1.2	96.3	2.5
Vertical	2.9	5.1	92.0

(a) Main Classes

	Solid	Porous	Object
Solid	73.8	13.0	13.2
Porous	3.4	89.2	7.4
Object	11.3	19.5	69.2

(b) Sub-vertical Classes

Table 4: Confusion matrices for main and sub-vertical classfication.

It is insightful to quantify to which extent temporal redundancy improves classification accuracy. To this end, we evaluate classification accuracy across different size of

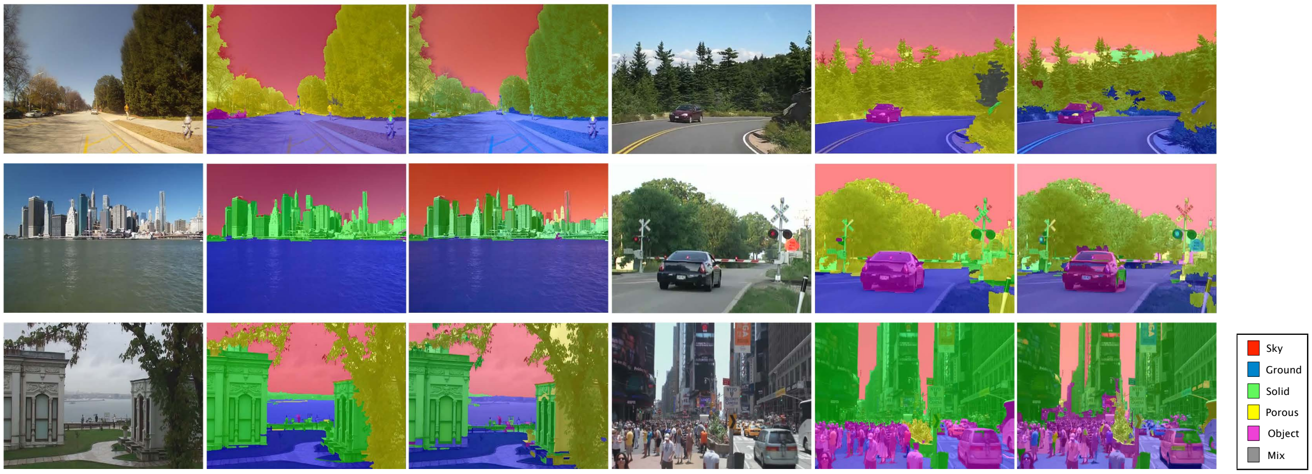
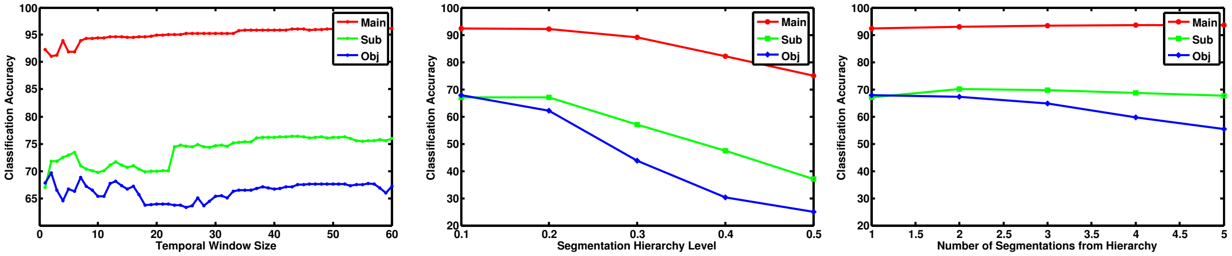


Figure 7: Qualitative results: From left to right: Input video frames, ground truth labels and predicted geometric labels. Our system performs well in challenging settings accurately predicting crowds, objects and foliage.



(a) Accuracy for different temporal windows.

(b) Classification results for various hierarchy levels.

Figure 8: (a) Temporal consistency improves accuracy for main and sub-vertical classifcation. The temporal window starts at the very first frame a segment appears in the video. (b) Classification accuracy estimated over 5-fold cross validation: (left) Single segmentation hierarchy level, (right) Multiple segmentation hierarchy levels. A temporal window size of one is used in both experiments.

temporal windows. Specifically, we compute the class-posteriors of a segment independently for each frame, obtaining the final probability by taking the average of the per-frame probabilities across the temporal window. Figure 8a shows the result for different lengths of temporal windows. It shows that accuracy reaches a stationary point for temporal windows of size 25 or greater. Using temporal window for labeling improves classification accuracy from 92.3% for a single frame to 96% for 25 frames for main classes, and from 67% to 77.4% for sub-vertical classes. However, accuracy for objects is virtually unchanged which we believe is due to the difficulty of segmenting these objects.

Figure 8b demonstrates the variation in classification accuracy when using a single versus multiple segmentation hierarchy levels. When using a single segmentation, the classification accuracy decreases with increasing hierarchy level from 0.1 to 0.5 (here 0.1 denotes the level at 10% of the overall hierarchy height). This decrease in accuracy is due to segments of different classes being increasingly mixed at higher hierarchy level as regions tend to get under segmented. Using multiple segmentations by combining different segmentation hierarchy levels provides a much more consistent accuracy, in particular it mitigates the problem of determining the correct granularity for a segment. In our experiments, combining predictions for geometric context at hierarchy levels 0.1 and 0.2 yields the best results.

Table 4 shows the row normalized confusion matrices. Notice, that we are able to achieve highly accurate classification results for main classes. For vertical sub-classes accuracy is lower, due to the vertical class containing huge intra-class variations and regions tend to be more affected by segmentation errors than the other classes. Finally, some qualitative miss-classifications are illustrated in Figure 10.

Importance of Features: We are using a wide variety of features covering appearance and motion. Here, we provide some insight into the importance of each individual feature type. To estimate the importance of a feature set, we only use the particular feature set across our cross-validation dataset to train and test our system. Table 5 shows the difference in accuracy when using only a particular feature set, here for a temporal window size of 1 frame. It can be seen, that the use of motion and appearance features yields the best accuracy, where motion features are primarily beneficial across the sub-vertical classifier (accuracy improves by

5% compared to appearance features alone). Table 5 also shows the benefit of temporal redundancy by using spatio-temporal regions. Compared to limiting features to only the very first frame for each region (last 2 rows in table),*i.e.* a setting similar to the image case, accuracy increases greatly (by a mean of 9% on the sub-vertical, and by 69.5% for the object class, in particular). This change is even more dramatic when comparing using all frames to using only the very first frame if limited to only appearance features (275%). Qualitative results are shown in Figure 9.

Features	*Main*	*Sub-Vertical*	*Object*
Motion & Appearance	92.3	67.0	67.8
Appearance only	92.3	64.0	64.7
Motion only	87.3	52.7	57.1
Motion & Appearance (first frame of segment only)	91.1	61.4	40.0
Appearance (first frame)	89.6	57.8	23.5

Table 5: Feature importance. We list the mean accuracy for the main and sub-vertical classifier and the individual accuracy of the object classifier. Using motion and appearance features yields the best accuracy (top row). Temporal redundancy is significant to our results, as shown by the reduced accuracy when limiting features to only the very first frame of each segment (last 2 rows).

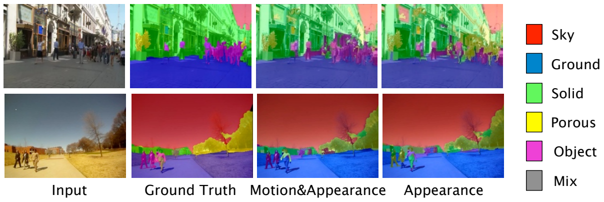


Figure 9: Qualitative comparison of importance of features. Overall, motion and appearance features achieve highest accuracy.

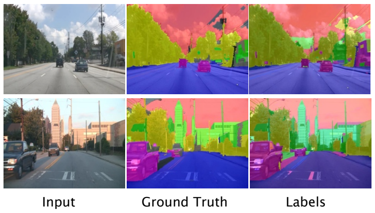


Figure 10: Misclassification examples: Scattered clouds are labeled as vertical class, a mix region of object / solid is labeled as car (top). Some ground is mistakenly labeled as object (bottom).

5.1. Semi-supervised Learning

Machine learning systems tend to improve performance with more training data available as intra-class variation is better accounted for. To verify this assumption for our dataset, we randomly pick 12, 24, 36, and 48 videos from the cross-validation dataset and restrict training to this set. Table 6a shows that accuracy indeed improves with increasing training data size, verifying our assumption.

Obtaining large amounts of training data requires manual annotation of videos, which is time consuming and requires crowd-based approaches [15] to scale. Alternatively, utilizing a large quantity of unlabeled data, we propose to adapt semi-supervised bootstrap learning. We iteratively train classifiers in a self-training manner, as shown in Figure 11. First classifiers (main, sub-vertical, and homogeneity) are trained using the annotated ground-truth data (1). Then, these classifiers are used to predict geometric context on unlabeled data (2). Segments with most confident labels (maximum class posterior $\geq 80\%$) are selected (3) and added to the training data with their predicted labels (4). In addition, we make use of multiple segmentations at different hierarchy levels, by including all high-confidence segments from the hierarchy that have high homogeneity (probability of being a single class $\geq 80\%$). Finally, the classifiers are re-trained on the expanded pool of labeled data and the process is iterated over. We expect accuracy on the added data to improve over several iterations. To avoid adding low quality segments to the labeled set, we perform introspection every 5th iteration, discarding added segments whose confidence (maximum class posterior) dropped below 80%.

No. of videos	*Main*	*Sub-Vertical*	*Object*
12	91.7	54.9	32.6
24	92.4	62.1	59.3
36	92.3	66.0	65.5
48	92.3	67.0	67.4

(a) Data-size dependency in supervised learning

Iteration	*Main*	*Sub-Vertical*	*Object*
0	85.1	74.7	73.0
5	85.2	74.2	75.0
10	86.2	77.2	79.9

(b) Semi-supervised bootstrap learning

Table 6: (a) Accuracy improves with larger training set size in supervised learning setting. (b) Leveraging semi-supervised learning (Figure 11) accuracy improves with successive iterations.

Table 6 demonstrates the effectiveness of our mutli-segmentation semi-supervised bootstrap learning. Our initial classifier is trained on a dataset of 63 videos (all videos in cross-valiation set, $\sim 200,000$ segments). At each iteration, we add 5,000 high confidence segments of each geometric class from unlabeled dataset, extending the training data. After 10 iterations, we are able to improve the performance by 1% for main, 3% for subvertical, and 9.5% for objects. In particular, we evaluate our bootstrap approach on a separate video dataset of 40 videos (7,000 frames). Comparing Table 6a and Table 6b shows that after 10 iterations we are able to achieve an improvement by semi-supervised bootstrap learning comparable to that of supervised learning.

6. Conclusion and Future Work

In this paper, we propose a novel algorithm for estimating geometric context in video, achieving highly accurate re-

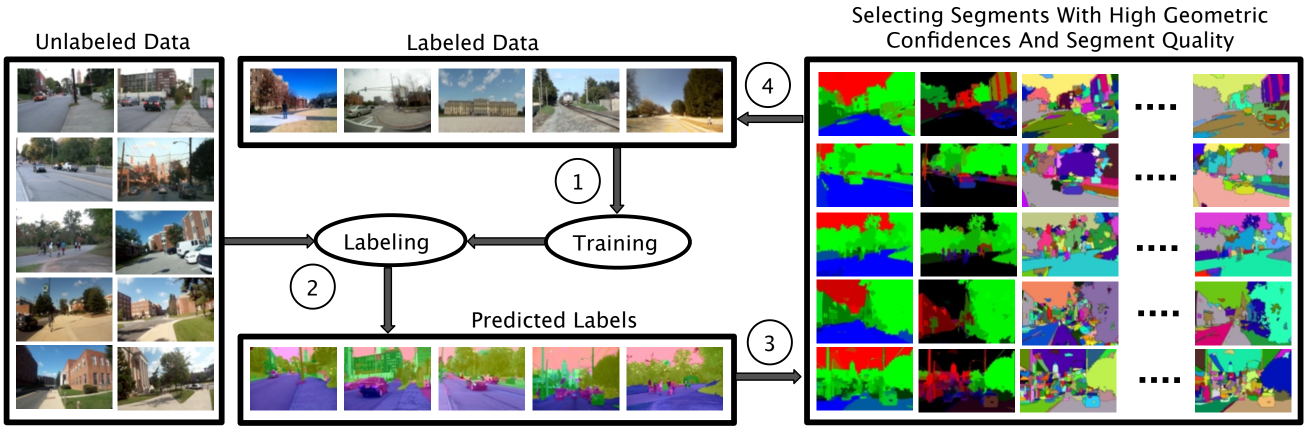

Unlabeled Data
Labeled Data
Selecting Segments With High Geometric Confidences And Segment Quality
④
①
Labeling
Training
②
Predicted Labels
③
....

Figure 11: Overview of semi-supervised bootstrap learning for geometric context in video (see Section 5.1). For segment selection, columns illustrate: (Left) confidence for main-classifier via color intenstiy (vertical: green, sky: red, ground: blue), (2nd) confidence for sub-vertical classifier (solid: red, porous: green, object: blue), (3rd and 4th) Segments across hierarchy levels (randomly colored).

sults. We thoroughly evaluate the contribution of motion features and demonstrated the benefit of utilizing temporal redundancy across frames. To measure accuracy of our approach, we collected a comprehensive dataset of annotated video which we plan to make available to the research community. We further showed how semi-supervised learning can broaden the pool of annotated data. To the best of our knowledge we demonstrate the first temporally consistent results for geometric context on *video*.

In the future, we plan to increase accuracy for the sub-vertical classifier. One reason for its lower accuracy is, that objects tend to be under-segmented even at the superpixel level, merging with porous or solid classes at higher hierarchy levels. We believe that improved segmentation of foreground objects will lead to enhanced accuracy of our method. Finally, we plan on leveraging geometric context to improve object detection and activity recognition in video.

Acknowledgement This material is based in part on research by the Defense Advanced Research Projects Agency (DARPA) under Contract No. W31P4Q-10-C-0214, and by a Google Grant and and Google PhD Fellowship for Matthias Grundman, who participated in this research as a Graduate Student at Georgia Tech. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any of the sponsors funding this research.

References

[1] G. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, 2008. 2

[2] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 2008. 2

[3] M. Collins, R.E. Schapire, and Y. Singer. Logistic regression, adaboost and bregman distances. *Machine Learning*, 2002. 3, 4

[4] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. *ECCV*, 2006. 4

[5] S.K. Divvala, D. Hoiem, J.H. Hays, A.A. Efros, and M. Hebert. An empirical study of context in object detection. In *IEEE CVPR*, 2009. 1

[6] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Scene parsing with multiscale feature learning, purity trees, and optimal covers. In *ICML*, 2012. 2

[7] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. *Image Analysis*, 2003. 4

[8] P.F. Felzenszwalb and D.P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004. 3

[9] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *IEEE CVPR*, 2009. 2

[10] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *IEEE CVPR*, 2010. 2, 3, 4

[11] D. Hoiem, A.A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, 2005. 2, 5

[12] D. Hoiem, A.A. Efros, and M. Hebert. Putting objects in perspective. In *IEEE CVPR*, 2006. 1

[13] D. Hoiem, A.A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 2007. 2, 4, 5

[14] O. Miksik, D. Munoz, J. A. Bagnell, and M. Hebert. Efficient temporal consistency for streaming video scene analysis. Technical report, RI, CMU, Sep. 2012. 2

[15] B.C. Russell, A. Torralba, K.P. Murphy, and W.T. Freeman. Labelme: a database and web-based tool for image annotation. *IJCV*, 2008. 7

[16] P. Sturgess, K. Alahari, L. Ladicky, and P. Torr. Combining appearance and structure from motion features for road scene understanding. In *BMVC*, 2009. 2

[17] J. Tighe and S. Lazebnik. Superparsing. *IJCV*, 2012. 2

[18] A. Torralba, K.P. Murphy, and W.T. Freeman. Contextual models for object detection using boosted random fields. In *NIPS*, 2004. 1

[19] C. Wojek, S. Roth, K. Schindler, and B. Schiele. Monocular 3d scene modeling and inference: Understanding multi-object traffic scenes. In *ECCV*. Springer, 2010. 2

[20] C. Xu and J.J. Corso. Evaluation of super-voxel methods for early video processing. In *IEEE CVPR*, 2012. 3