Pattern Recognition Letters 30 (2009) 88–97

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Semantic object classes in video: A high-definition ground truth database

Gabriel J. Brostow [a,b,*], Julien Fauqueur [a], Roberto Cipolla [a]

[a] Computer Vision Group, University of Cambridge, United Kingdom
[b] Computer Vision and Geometry Group, ETH Zurich

ARTICLE INFO

Article history:
Available online 22 April 2008

PACS:
87.57.N–
42.30.Tz
87.57.nm
42.30.Sy

Keywords:
Object recognition
Video database
Video understanding
Semantic segmentation
Label propagation

ABSTRACT

Visual object analysis researchers are increasingly experimenting with video, because it is expected that motion cues should help with detection, recognition, and other analysis tasks. This paper presents the Cambridge-driving Labeled Video Database (CamVid) as the first collection of videos with object class semantic labels, complete with metadata. The database provides ground truth labels that associate each pixel with one of 32 semantic classes.

The database addresses the need for experimental data to quantitatively evaluate emerging algorithms. While most videos are filmed with fixed-position CCTV-style cameras, our data was captured from the perspective of a driving automobile. The driving scenario increases the number and heterogeneity of the observed object classes. Over 10 min of high quality 30 Hz footage is being provided, with corresponding semantically labeled images at 1 Hz and in part, 15 Hz.

The CamVid Database offers four contributions that are relevant to object analysis researchers. First, the *per-pixel* semantic segmentation of over 700 images was specified manually, and was then inspected and confirmed by a second person for accuracy. Second, the high-quality and large resolution color video images in the database represent valuable extended duration digitized footage to those interested in driving scenarios or ego-motion. Third, we filmed calibration sequences for the camera color response and intrinsics, and computed a 3D camera pose for each frame in the sequences. Finally, in support of expanding this or other databases, we present custom-made labeling software for assisting users who wish to paint precise class-labels for other images and videos. We evaluate the relevance of the database by measuring the performance of an algorithm from each of three distinct domains: multi-class object recognition, pedestrian detection, and label propagation.

1. Introduction

Training and rigorous evaluation of video-based object analysis algorithms require data that is labeled with ground truth. Video labeled with semantic object classes has two important uses. First, it can be used to train new algorithms that leverage motion cues for recognition, detection, and segmentation. Second, such labeled video can be useful to finally evaluate existing video algorithms *quantitatively*.

This paper presents the CamVid Database, which is to our knowledge, the only currently available video-based database with per-pixel ground truth for multiple classes. It consists of the original high-definition (HD) video footage and 10 min of frames which volunteers hand-labeled according to a list of 32 object classes. The pixel precision of the object labeling in the frames allows for accurate training and quantitative evaluation of algorithms. The database also includes the camera pose and calibration parameters of the original sequences. Further, we propose the InteractLabeler, an interactive software system to assist users with the manual labeling task. The volunteers' paint strokes were logged by this software and are also included with the database.

We agree with the authors of Yao et al. (2007) that perhaps in addition to pixel-wise class labels, the semantic regions should be annotated with their shape or structure, or perhaps also organized hierarchically. Our data does not contain such information, but we propose that it may be possible to develop a form of high-level boundary-detection in the future that would convert this and other pixel-wise segmented data into a more useful form.

1.1. Related work

So far, modern databases have featured still images, to emphasize the breadth of object appearance. Object analysis algorithms are gradually maturing to the point where scenes (Lazebnik et al., 2006; Oliva and Torralba, 2001), landmarks (Snavely et al.,

* Corresponding author. Address: Computer Vision Group, University of Cambridge, United Kingdom. Fax: +44 1223 332662.
E-mail address: brostow@cc.gatech.edu (G.J. Brostow).
URL: http://mi.eng.cam.ac.uk/research/projects/VideoRec/ (G.J. Brostow).

0167-8655/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2008.04.005

Dataset	Classes	# Labeled Frames	Annotation
Berkeley Segmentation (Martin et al., 2001)	NA	300 stills	Object edges
Caltech 101 (Fei-Fei et al., 2006)	101	5000 stills	Polygons
Caltech 256 (Griffin et al., 2007)	256	30607 stills	Polygons
CBCL StreetScenes (Bileschi, 2006)	9	3547 stills	Polygons
LabelMe (Russell et al.,2005)	183	41996 stills	Polygons
VOC 2007 Classif. (PASCAL Visual Object Classes Challenge)	20	5011 stills	Bounding boxes
VOC 2007 Segment. (PASCAL Visual Object Classes Challenge)	20	422 stills	Pixel-wise masks
Imageparsing.com (Yao et al., 2007)	220	25449 stills	Pixel-wise masks
MSR Cambridge v2 (Shotten et al., 2006)	23	591 stills	Pixel-wise masks
Ours	32	701 of 10min video	Pixel-wise masks
PETS 2007	3 events	17.7min video	Time indexes
TRECVid 2007 (Smeaton et al., 2006)	36	120 hrs.	Per-shot presence

Fig. 1. This table compares the most relevant and now publicly available image and video databases. These datasets can variously be used for multi-class object recognition, detection, event analysis, and semantic segmentation. Our database mainly targets research on pixel-wise semantic segmentation (so segmentation and recognition). The Berkeley set is a small subset of the previously popular 800 Corel Photo CDs (Müller et al., 2002), and is listed here because their hand-marked edges are spatial annotations that are publicly available and potentially useful. LabelMe has further hand-typed class names, but 183 with more than 30 occurrences. Imageparsing.com (Yao et al., 2007) has unknown numbers of its many classes, and is said to still be growing. Except for PETS which is 768 × 576, most data is VGA (640 × 480) or smaller, while ours is HD (960 × 720). *Ours* and two other datasets consist of video, and those two provide little spatial context annotation: PETS lists the time and quadrant when one of three events occurs (loitering, theft, unattended baggage), and TRECVid flags when a clip contains one of the 36 footage types (sports, weather, court, office, etc.). Of the databases with pixel-wise class segmentation, ours has the most labeled frames, and our frames have the fewest unlabeled pixels. (See above-mentioned references for further information.)

2006), and whole object classes (Rabinovich et al., in press) could be recognized in still images for a majority of the test data (Fei-Fei et al., 2006).

We anticipate that the greatest future innovations in object analysis will come from algorithms that take advantage of spatial and temporal context. Spatial context has already proven very valuable, as show by Hoiem et al. (2006) who took particular advantage of perspective cues. Yuan et al. (2007) showed the significant value of layout context and region adaptive grids in particular. Yuan et al. experimentally demonstrated improved performance for region annotation of objects from a subset of the Corel Stock Photo CDs which they had to annotate themselves for lack of existing labels. They have a lexicon of 11 concepts that overlaps with our 32 classes. Our database is meant to enable similar innovations, but also for temporal context instead of spatial. Fig. 1 lists the most relevant photo and video databases used for either recognition or segmentation.

The performance of dedicated detectors for cars (Leibe et al., 2007) and pedestrians (Dalal and Triggs, 2005) is generally quantified thanks to data where individual entities have been counted, or by measuring the overlap with annotated bounding boxes. A number of excellent still-image databases have become available recently, with varying amounts of annotation. The Microsoft Research Cambridge database (Shotton et al., 2006) is among the most relevant, because it includes per-pixel class labels for every photograph in the set. The LabelMe (Russell et al., 2005) effort has cleverly leveraged the internet and interest in annotated images to gradually grow their database of polygon outlines that approximate object boundaries. The PASCAL Visual Object Classes Challenge provides datasets and also invites authors to submit and compare the results of their respective object classification (and now segmentation) algorithms.

However, no equivalent initiative exists for video. It is reasonable to expect that the still-frame algorithms would perform similarly on frames sampled from video. However, to test this hypothesis, we were unable to find suitable existing video data with ground truth semantic labeling.

In the context of video based object analysis, many advanced techniques have been proposed for object segmentation (Marcotegui et al., 1999; Deng and Manjunath, 2001; Patras et al., 2003; Wang et al., 2005; Agarwala et al., 2004). However, the numerical evaluation of these techniques is often missing or limited. The results of video segmentation algorithms are usually illustrated by a few segmentation examples, without quantitative evaluation. Interestingly, for detection in security and criminal events, the PETS Workshop (The PETS, 2007) provides benchmark data consisting of event logs (for three types of events) and boxes. TRECVid (Smeaton et al., 2006) is one of the reigning *event analysis* datasets, containing shot-boundary information, and flags when a given shot "features" sports, weather, studio, outdoor, etc. events.

1.2. Database description

We propose this new ground truth database to allow numerical evaluation of various recognition, detection, and segmentation techniques. The proposed CamVid Database consists of the following elements:

- the original video sequences (Section 2.1);
- the intrinsic calibration (Section 2.2);
- the camera pose trajectories (Section 2.3);
- the list of class labels and pseudo-colors (Section 3);
- the hand labeled frames (Section 3.1);
- the stroke logs for the hand labeled frames (Section 3.2).

The database and the InteractLabeler (Section 4.2) software shall be available for download from the web.[1] A short video provides an overview of the database. It is available as supplemental material to this article, as well as on the database page itself.

[1] http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/.

2. High quality video

We drove with a camera mounted inside a car and filmed over two hours of video footage. The CamVid Database presented here is the resulting subset, lasting 22 min, 14 s. A high-definition 3CCD Panasonic HVX200 digital camera was used, capturing 960 × 720 pixel frames at 30 fps (frames per second). Note the pixel aspect ratio on the camera is not square and was kept as such to avoid interpolation and quality degradation.[2] The video files are provided straight from the camera with native DVC-PRO HD compression, 4:2:2 color sampling and non-interlaced scan. We provide a small custom program that extracts PNG image files with appropriate file names. As shown in Fig. 2, the camera was set up on the dashboard of a car, with a similar field of view as that of the driver.

2.1. Video sequences

The CamVid database includes four HD video sequences. Their names and durations (in minutes:seconds) are: `0001TP` (8:16), `0006R0` (3:59), `0016E5` (6:19), and `Seq05VD` (3:40). The total duration is then 22:14.

The sequence `0001TP` was shot at dusk. Objects in the scene can still be identified, but are significantly darker and grainier than in the other sequences. Vehicles did not yet have their lights on. The other three sequences were captured in daylight in rather sunny weather conditions. The environment is mixed urban and residential.

The three daytime sequences were shot and selected because they contain a variety of important class instances:

- cars;
- pedestrians;
- cyclists;

and events:

- moving and stationary cars;
- cyclists ahead and along side the car;
- pedestrians crossing;
- driving through a supermarket parking lot;
- accelerating and decelerating;
- left and right turns;
- navigating roundabouts.

The high definition of these videos is necessary to allow future algorithms to detect small objects, such as traffic lights, car lights, road markings, and distant traffic signs.

2.2. Camera calibration

The CamVid Database is fairly unique in that it was recorded with a digital film camera which was itself under controlled conditions. This means that no zooming, focus, or gain adjustments were made during each sequence. Focus was set at infinity, and the gain and shutter speed were locked. The aperture was opened at the start as much as possible without allowing white objects in the scene to saturate.

Anticipating that the CamVid Database may someday be used for photometric experiments, we filmed the standard Gretag–Macbeth color chart, shown in Fig. 3A. We captured it at multiple angles, so that others will be able to factor out the secondary lighting bouncing from buildings and other geometry.

[2] Most HD televisions obtain square pixels by resizing such video to 1280 × 720.

Fig. 2. High-definition camera mounted on the dashboard.

To aid with projective reconstruction, the camera's intrinsics were also calibrated. We filmed a calibration pattern at multiple angles (Fig. 3B) and followed the calibration procedure of Bouguet (Bouguet, 2004). The database contains both the images and our estimated calibration. In the process, we computed focal lengths (f_x, f_y), the principal point (a_x, a_y), and the lens distortion parameters r. The database is presented with the footage in its raw form, so pixel columns have not been interpolated as they would be for HD broadcast, and no lens-distortion correction has been applied to correct for r.

2.3. Camera pose tracking

Moving objects appear in most frames of the database, but static objects are substantially larger in terms of pixel area. Consequently, ego-motion dominates the sequences, except where the car came to a stop. The camera was tracked for 3D reconstruction purposes, and to help users of the database factor out appearance changes due to ego-motion. We used the industry-standard matchmoving software Boujou 3.0 (Boujou, 2007) to compute the camera's 3D pose in each frame.

Each continuous driving segment lasted several minutes, producing more frames than the matchmoving software could handle at once. We wrote scripts to process 1000 frames at a time, with 50 frames of overlap on each successive subsequence. These, in turn, were processed to track 2D features by Boujou. Their robust estimator selected a subset of these feature trajectories as inliers, and computed the sequence's pose over time. Consequently, the CamVid Database contains the following ascii data files for each of the four segments (broken up into 1000 frame subsequences)

- sparse position matrix of all 2D features over all frames,
- per-frame rotation matrix R and translation vector t of the camera,
- a subset of features that were inliers to the pose-estimation, resulting in 3D world coordinates X.

Fortunately, none of the footage contained frames where a large vehicle filled the field of view, so the dominant motion was always due only to the car.

3. Semantic classes and labeled data

After surveying the greater set of videos, we identified 32 classes of interest to drivers. The class names and their corresponding pseudo-colors are given in Fig. 4. They include fixed objects, types of road surface, moving objects (including vehicles and people),

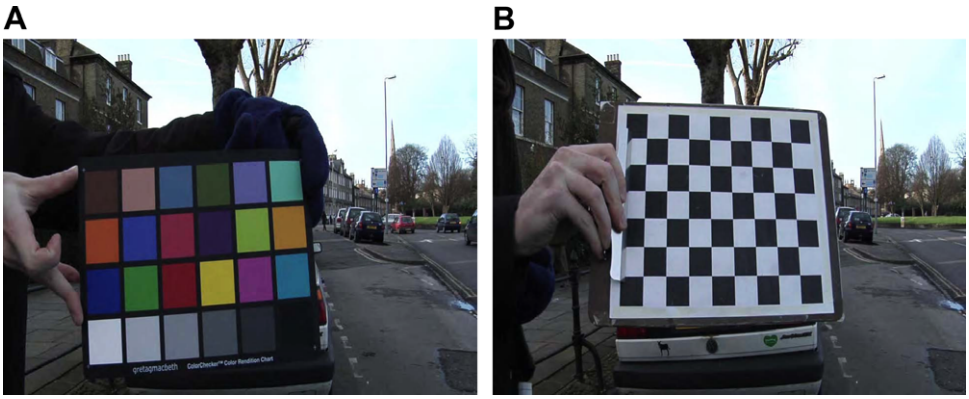

Fig. 3. (A) Sample image from a color calibration sequence featuring the Gretag–Macbeth color chart. (B) Sample image of the checkerboard pattern used to estimate intrinsic calibration parameters.

Fig. 4. List of the 32 object class names and their corresponding colors used for labeling.

and ceiling (sky, tunnel, archway). The relatively large number of classes (32) implies that labeled frames provide a rich semantic description of the scene from which spatial relationships and context can be learned.

The correspondence between class names and (R, G, B) color values used for labeling are given in a file as part of the database. Note the special status of the `Void` label: it indicates an area which is semantically ambiguous or irrelevant in this context. We re-used the color-index assignments from Shotton et al. (2006), and added the new classes according to their indexing scheme.

3.1. Labeled ground truth frames

Seven hundered and one frames from the video sequences were manually labeled using the color indexing given in Fig. 4, and the tools described in Section 4. This represents approximately 230 man-hours of labeling.

As opposed to bounding boxes or approximate boundaries, the pixel-precision of the class labeling shown in Fig. 5 could simplify accurate learning of appearance and shape.

The pixel-precision of the class labeling (as opposed to bounding boxes or approximate boundaries) allows for accurate learning of appearance and shape, as shown in Fig. 5.

Fig. 6 describes the five different labeled frame sequences available. They were extracted from the four original video sequences at a rate of 1 fps, by considering every 30th frame (the recording frame rate was 30 fps). However the `0016E5_15Hz` sequence was also extracted at 15 fps, by considering every other frame. It corresponds to the `CamSeq01` dataset used in (Fauqueur et al., 2007) for the problem of automatic label Propagation. This faster temporal sampling allows for finer algorithm evaluation. The last column of Fig. 6 indicates the corresponding duration of the video sequences, calculated as the product of the frame rate and number of labeled frames. The overall duration of those labeled sequences is about 10 min. Within these labeled 10 min, the variety of objects and events constitute a rich ground truth.

Note that although the labeled sequences have sub realtime frame rates (1 or 15 fps), their corresponding original video sequences are at 30 fps. Algorithms can be run on the corresponding original sequences. Such higher frame rates may be required for algorithms that depend on temporal smoothness, such as optical flow computation. The evaluations can then be run at 1 or 15 Hz using our ground truth.

In Fig. 7, we report the label statistics of those sequences according to the 32 classes. The statistics indicate the number of frames and the number of instances per object class that this ground truth provides. Column "%" provides the proportion of pixels belonging to each class across each sequence, while "occ." refers to the number of frames in which a class instance occurs *i.e.*, when at least one pixel has the label. All six tables are sorted by decreasing order of "occ." then "%". The first table reports the statistics when all five sequences are considered together and hence gives an overview.

In the first table, we observe that only 2.68% of pixels overall were labeled as `Void`, *i.e.*, not assigned to a class. This low rate is an indicator of consistent labeling and also shows that the choice of the 32 class labels was adequate for this problem.

Since `Road`, `Building` and `Sky` classes constitute the urban setting, we consistently note they represent the largest classes (between 15.81% and 27.35%) and they are present in virtually all frames. On contrary, `Car` and `Pedestrian` classes, which are crucial entities in our context are very frequent (high "occ." value) but

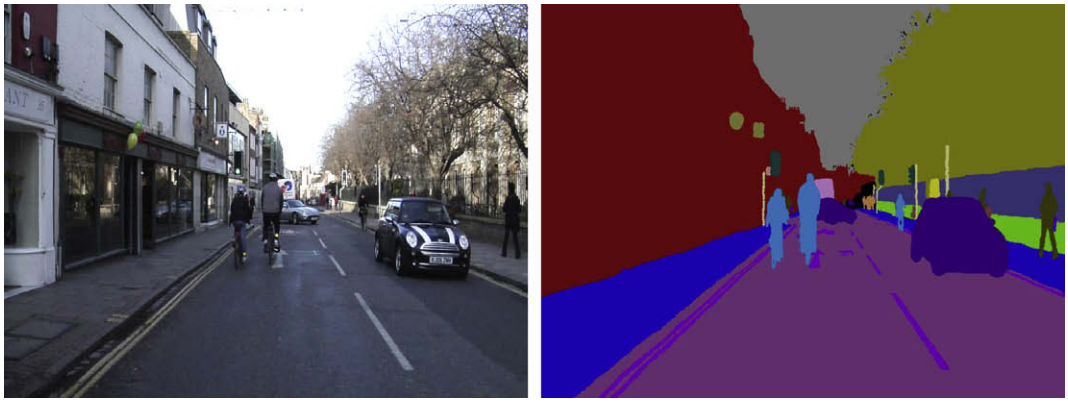

Fig. 5. Example of a captured frame and its corresponding labeled frame. Refer to Fig. 4 for the label correspondences.

labeled sequence name	original sequence name	frame rate in fps	number of labeled frames	corresponding duration
0001TP_L	0001TP	1	124	2:04
0016E5_1Hz_L	0016E5	1	204	3:24
0016E5_15Hz_L	0016E5	15	101	0:06
0006R0_L	0006R0	1	101	1:41
Seq05VD_L	Seq05VD	1	171	2:51
	Total		701	10:06

Fig. 6. Naming and lengths of the five labeled sequences.

small (3.97% and 0.64%). Other entities which are critical (such as `Child` and `RoadShoulder`) are also important and small but also rare. From a statistical learning point of view, these tables tell us which object classes are rare, hence difficult to learn. They also show the degree of imbalance between the different classes which may be considered in a learning algorithm.

Finally, the relative importance of those classes can vary according to the location. For example, the sequence `0006R0` has fewer `Building` pixels and more `Tree` pixels than the other sequences, because it was shot in a less developed area.

In summary, the key aspects of the ground truth annotations are: the pixel resolution labeling, the few unlabeled pixels, the high image resolution, the semantic descriptions, and the extended duration (about 10 min).

3.2. Paint stroke logs

Our labeling software (Section 4.2) logged the paint strokes of volunteers while they labeled the 701 frames. These stroke logs contain all the user actions, so they can be used to replay the labeling process for each frame. For each semantic paint stroke, the following information is logged: the selected semantic label color (*i.e.*, class), the brush radius, the 2D path of stroke points, segmentation parameters (*i.e.*, which edges bounded the flood-filling), time stamps, and the user-time taken to paint each stroke. The logs show that while some classes are especially easy to label, the annotation of others would benefit from further automated assistance.

4. Production of the labeled frames data

For 701 frames extracted from the database sequence, we hired 13 volunteers (the "labelers") to manually produce the corresponding labeled images. They painted the areas corresponding to a predefined list of 32 object classes of interest, given a specific palette of colors (Fig. 4).

In this section, we give an overview of the website (Section 4.1) and the labeling software (Section 4.2) that were designed for this task. The website has allowed us to train volunteers and then exchange original and labeled frames with them for the past seven months. The software is the tool used by the labelers to produce the labeled frames.

4.1. Website for volunteers

We built a website written in `php` to allow the labelers to log in, download the labeling software, download new frames to label, upload the corresponding labeled frames, and keep track of their progress. To recruit non-expert labelers in the first place, potential volunteers were directed to our website by our paid advertisements on FaceBook (Facebook homepage, 2007), a popular online social network site. Of 89 initial volunteers, 13 have submitted acceptable quality labeled images. A table contains the status of each frame to indicate whether it is (i) unassigned to a labeler, (ii) assigned but not labeled, (iii) assigned and labeled or (iv) approved. The approved-status means that we had manually inspected the labeled frame to make sure that no object was left out and that the labeling style, which potentially varies depending on the person, is consistent across the dataset. If the labeling quality was too poor, we ask the labeler to re-label it, otherwise we would refine the labeling, if necessary, and approve it. For the time being, each labeler is paid for each approved frame.

On this website, we have also provided instructions to encourage a consistent labeling:

- "Getting the labels right": gives a description of the meaning of each class label.
- "Avoid holes": requires that holes and cracks possibly left by the segmentation algorithms be filled.
- "Precision of object boundaries": asks labelers to paint along object boundaries as accurately as possible.
- "All obvious and clearly visible objects should be assigned a label."

4.2. InteractLabeler: object labeling tool

We developed InteractLabeler, a software program, to assist the labelers in the painting task, by providing a suitable GUI and offering different automatic pre-segmentations. The source-code is in C++ using The OpenCV Library. The program and detailed instructions are on the project web-page. We received feedback and

All 5 sequences all together
701 frames

class name	%	occ.
Road	27.3	701
Void	3.12	701
Sky	15.8	699
Column_Pole	0.98	698
LaneMkgsDriv	1.7	696
Building	22.7	687
Sidewalk	6.33	672
Car	3.4	643
Pedestrian	0.64	640
Tree	10.4	636
Misc_Text	0.58	582
TrafficLight	0.37	471
Wall	1.35	469
OtherMoving	0.39	422
Sign_Symbol	0.12	416
Bicyclist	0.53	365
Fence	1.43	363
SUVPickupTruck	0.7	304
VegetationMisc	0.75	265
CartLuggagePram	0.03	243
ParkingBlock	0.33	189
Truck_Bus	0.54	183
Child	0.03	142
Archway	0.06	114
RoadShoulder	0.26	55
Animal	0	26
LaneMkgsNonDriv	0.02	21
TrafficCone	0	20
MotorcycleScooter	0.01	14
Bridge	0.05	10
Tunnel	0.00	1
Train	0	0

Sequence 0001TP
124 frames

class name	%	occ.
Building	19.6	124
Road	16.1	124
Tree	15.8	124
Void	6.36	124
Car	5.96	124
Sidewalk	4.39	124
Sky	20.1	123
LaneMkgsDriv	0.85	123
Column_Pole	0.81	123
Pedestrian	0.88	119
SUVPickupTruck	1.5	118
TrafficLight	0.31	107
Misc_Text	0.48	102
Wall	2.16	90
Bicyclist	0.51	75
OtherMoving	0.16	67
CartLuggagePram	0.04	59
VegetationMisc	0.29	47
Truck_Bus	2.4	43
Fence	0.6	41
Sign_Symbol	0.03	39
ParkingBlock	0.27	29
RoadShoulder	0.21	12
Tunnel	0	0
Train	0	0
TrafficCone	0	0
MotorcycleScooter	0	0
LaneMkgsNonDriv	0	0
Child	0	0
Bridge	0	0
Archway	0	0
Animal	0	0

Sequence 0006R0
101 frames

class name	%	occ.
Road	33.7	101
Sky	20.3	101
Void	2.14	101
Column_Pole	1.05	101
Tree	15	97
LaneMkgsDriv	1.8	97
Car	5.24	93
Building	10.6	90
Misc_Text	1.63	85
Sidewalk	1.63	72
Pedestrian	0.33	68
SUVPickupTruck	0.81	66
VegetationMisc	2.64	64
OtherMoving	0.79	64
Fence	0.77	44
Sign_Symbol	0.07	37
ParkingBlock	0.72	31
CartLuggagePram	0.02	27
Wall	0.19	25
TrafficCone	0.02	16
Archway	0.31	15
MotorcycleScooter	0.07	14
Bridge	0.1	8
TrafficLight	0.01	7
Animal	0.00	1
Tunnel	0	0
Truck_Bus	0	0
Train	0	0
RoadShoulder	0	0
LaneMkgsNonDriv	0	0
Child	0	0
Bicyclist	0	0

Sequence 0016E5_15Hz
101 frames

class name	%	occ.
Road	27.1	101
Building	24.2	101
Tree	16.3	101
Sky	9.17	101
Sidewalk	8.59	101
Bicyclist	2.23	101
LaneMkgsDriv	1.71	101
Void	1.58	101
Pedestrian	0.67	101
TrafficLight	0.59	101
Fence	3.07	100
Wall	1.43	100
Column_Pole	0.51	100
Truck_Bus	0.3	90
OtherMoving	0.67	88
Sign_Symbol	0.07	82
Child	0.08	81
Archway	0.03	74
Car	1.46	62
Misc_Text	0.17	61
CartLuggagePram	0.01	37
VegetationMisc	0.04	20
LaneMkgsNonDriv	0.01	1
Tunnel	0	0
Train	0	0
TrafficCone	0	0
SUVPickupTruck	0	0
RoadShoulder	0	0
ParkingBlock	0	0
MotorcycleScooter	0	0
Bridge	0	0
Animal	0	0

Sequence 0016E5_1Hz
204 frames

class name	%	occ.
Road	31.2	204
Building	27.4	204
Sidewalk	4.99	204
Void	2.96	204
LaneMkgsDriv	1.75	204
Sky	14.6	203
Car	3.61	203
Column_Pole	0.98	203
Misc_Text	0.51	191
Pedestrian	0.78	190
Tree	5.12	159
Bicyclist	0.33	140
TrafficLight	0.33	132
Sign_Symbol	0.08	117
Fence	1.52	116
OtherMoving	0.3	115
Wall	0.54	110
SUVPickupTruck	1.04	97
CartLuggagePram	0.05	92
ParkingBlock	0.58	89
VegetationMisc	0.29	69
Truck_Bus	0.24	50
Child	0.05	38
RoadShoulder	0.63	31
Archway	0.05	13
LaneMkgsNonDriv	0.02	6
Animal	0	3
Bridge	0.1	2
TrafficCone	0.00	1
Tunnel	0	0
Train	0	0
MotorcycleScooter	0	0

Sequence Seq05VD
171 frames

class name	%	occ.
Road	26.9	171
Sky	15.5	171
Sidewalk	10.8	171
Void	2.44	171
LaneMkgsDriv	2.18	171
Column_Pole	1.34	171
Building	25.6	168
Pedestrian	0.47	162
Car	1.37	161
Tree	6.75	155
Wall	2.38	144
Misc_Text	0.34	143
Sign_Symbol	0.29	141
TrafficLight	0.56	124
OtherMoving	0.28	88
VegetationMisc	0.94	65
Fence	1.33	62
Bicyclist	0.07	49
ParkingBlock	0.05	40
CartLuggagePram	0.01	28
SUVPickupTruck	0.07	23
Child	0.01	23
Animal	0.02	22
LaneMkgsNonDriv	0.05	14
RoadShoulder	0.15	12
Archway	0	12
TrafficCone	0	3
Tunnel	0.00	1
Truck_Bus	0	0
Train	0	0
MotorcycleScooter	0	0
Bridge	0	0

Fig. 7. Statistics for each class through each labeled sequence: "%" indicates the proportion of pixels and "occ." the number of occurrences. Rows are sorted by decreasing number of occurrences then decreasing "%".

suggestions from labelers, that helped us improve its usability over three release versions.

Imprecise edges had been the norm with the plain brush tools because labelers would paint at a safe distance from edges, knowing that assigning pixels (on the far side of an edge) to the wrong class would cause serious errors for learning algorithms. Conversely, the classic bounding box alternative to object labeling is also likely to cause learning

errors since background pixels are labeled as object pixels.

With the new InteractLabeler, the labeler can toggle between three different windows for each frame of the sequence, as shown in Fig. 8: the original frame, the labeled map being painted, and a zooming window. A console window gives additional information about the selected labels and the segmentation algorithms. The label color selection table (see Fig. 4) can be displayed in the original frame window.

When a frame is loaded, three automatic segmentation algorithms are run: Felzenszwalb and Huttenlocher (2004), Mean Shift (Comaniciu and Meer, 1997) and the pyramid-based algorithm of Burt et al. (1981), as implemented in The OpenCV Library. When an object has to be painted with the selected color of its class, the three segmentation results can be overlaid in the labeled map window. The segmented region that best overlaps with the object can be floodfilled with the class color when the user clicks or paints a stroke with the mouse-cursor. Most pixels are painted in this manner. It is sometimes necessary to manually refine the labeling in highly textured or saturated (due to specular reflection) areas. These refinements are performed in the same manner, but with a free-hand brush tool. The brush radius can be easily adjusted to match the area to be painted. A small brush will be chosen to accurately paint small areas in a zoomed in view of the frame. The zooming factor is adjusted interactively using the mouse-wheel.

In this way, object recognition algorithms that rely on clean and accurate contours are less likely to face imprecisely labeled edges. Other features of InteractLabeler include various shortcut keys (to undo/redo, navigate through the sequence, toggle through the segmentation modes) and the possibility to adjust the transparency of the overlaid original frame in the labeled map.

Another important aspect of this tool is its built-in logging of the user strokes. As described in Section 3.2, the log file for each frame contains a full description of the user actions. By logging and timing each stroke, we were able to estimate the hand labeling time for one frame to be around 20–25 min. This duration can vary greatly depending on the complexity of the scene. The stroke logs are part of the database.

InteractLabeler is available along with the CamVid Database. It can be used to expand the ground truth database by hand labeling more frames from the sequences, in addition to those already provided (Section 3.1). Beyond the application of automated driving vehicles, it can also be used to either label image sequences or video files for any other domain: a text file of class names and corresponding colors is given as an argument to the program. Other researchers may find this software valuable to produce ground truth for their own data and applications relating to object recognition, tracking, and segmentation.

5. Applications and results

To evaluate the potential benefits of the CamVid Database, we measured the performance of several existing algorithms. Unlike many databases which were collected for a single application, the CamVid Database was intentionally designed for use in multiple domains. Three performance experiments examine the usefulness of the database for quantitative algorithm testing. The algorithms address, in turn, (i) object recognition, (ii) pedestrian detection, and (iii) segmented label propagation in video. The challenges addressed by the first two algorithms have classically ignored video (Dalal et al. (2006) is a notable exception), while algorithms from the third class have typically been tested on short and contrived indoor videos (Piroddi and Vlachos, 2002).

5.1. Object recognition

Recognition of objects in natural scenes is one of the defining challenges of computer vision. The most advanced algorithms still compete with each other using training and test data consisting of

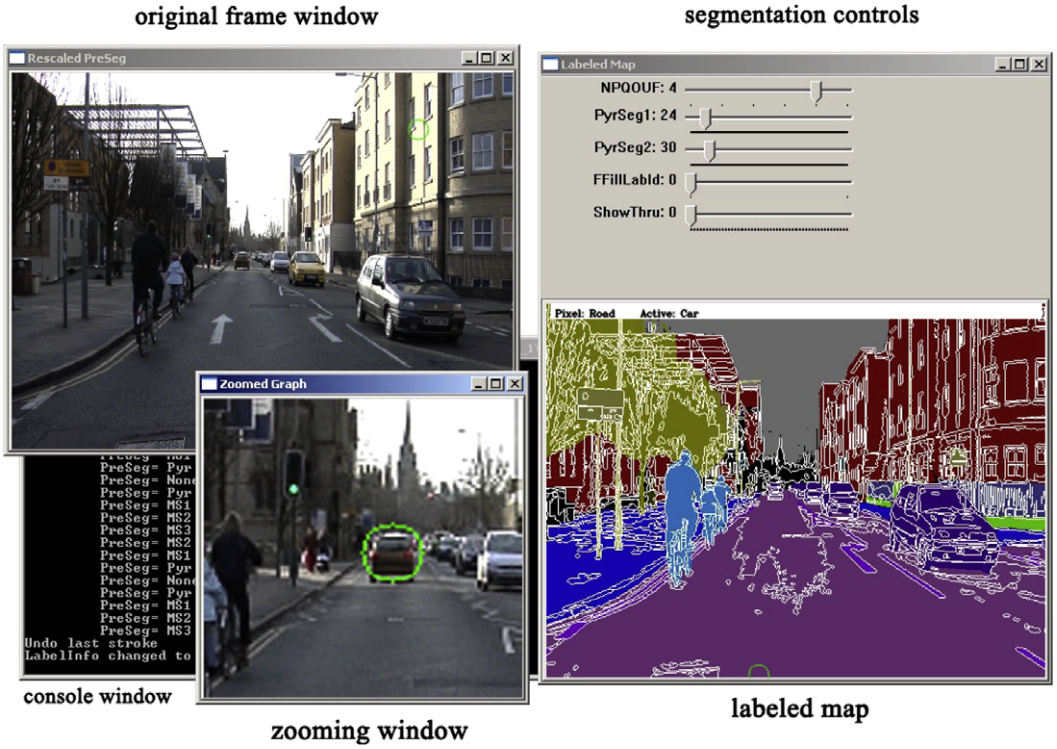

Fig. 8. Graphic user interface of InteractLabeler labeling software used by volunteers to assign a semantic label to *each* pixel. The combination of various pre-segmentations, zooming and manual refinement capabilities assist the user in the precise labeling task.

only static images (PASCAL Visual Object Classes Challenge). They should compete on videos because: (1) the extra data could make recognition easier, and (2) most real-world applications deal with video *e.g.*, CCTV cameras and modern news-archiving.

State of the art algorithms such as TextonBoost (Shotton et al., 2006; Rabinovich et al., in press) are very dependent on the quality and quantity of available training data. For these algorithms to succeed, the data must contain per-pixel semantic labels, and is naturally in short supply. Section 5.3 and our work in (Fauqueur et al., 2007) specifically target the propagation of existing labels from one frame of a video when video is available. To start, our labeled images are worth evaluating in the context of TextonBoost, to show that the data is compatible with existing object recognition techniques.

Processing the CamVid Database using TextonBoost gave very similar overall scores to those previously obtained on the MSR-Cambridge data. TextonBoost works by building up a strong classifier in the form of a chain of weak classifiers. Each successive weak classifier stump is selected to maximally reduce the training error of the classifier-chain so far. Rather than pixel intensities directly, the image features being classified are grouped as 200 texton (Leung and Malik, 2001) clusters. Possible weak classifiers are then drawn from a pool of 40,000 combinations of a 2D offset, a rectangle size, and a texton cluster ID. The TextonBoost paper (Shotton et al., 2006) contains further implementation details.

The reported overall pixel-wise accuracy for the 23-class MSR-Cambridge database was a 72.2% recognition rate on 9 classes. That score is comparable to the daytime 73.5% recognition rate we achieved on our CamVid Database, which we grouped into 11 larger classes for this experiment to better reflect the statistically significant classes (see Fig. 7).

Using our database, we observed an opportunity to slightly improve the TextonBoost algorithm. The threshold stumps for the texton integral images were spaced equally between 0 and 1000 in the original algorithm. We now find the real range automatically, and normalize, insuring that the weak learners correctly span the full range. Consequently, the daytime performance increased further from 73.5% to 75.02%. A per-class graph of TextonBoost results (on daytime footage) is shown as orange[3] bars in Fig. 9. Blue bars plot performance of training and testing on the dusk parts of the database. TextonBoost is known to be a memory-hungry algorithm, which has meant that all tests required that footage from the database be resized to 320×240 pixels and further subsampled. We speculate that removing this restriction would improve scores for some of the smaller classes.

The result of processing footage with the same distribution of classes but more challenging appearance is largely worse (see the blue bars in Fig. 9). It would be unreasonable to expect vision algorithms to function in the dark, but the fact is that humans were able to recognize classes in the more challenging dusk sequence. Overall, the CamVid Database is shown to contain footage that both matches the assumptions of current algorithms, and sets a high bar on performance for future techniques. We suggest that future work could take even more advantage of the available temporal information.

5.2. Pedestrian detection

Pedestrian detection is another important objective in our research community, and is especially important for security and automated driving applications. Dalal and Triggs's (2005) is currently one of the leading detection algorithms and has the benefit of being multi-scale. The algorithm works by performing local contrast normalization of the images, then computing Histograms of Oriented Gradients (HOGs) on a coarse local grid. A linear SVM classifier is trained using positive and negative example sub-images of humans. Finally, the classifier is retrained with the false-positives added to the existing training data to tighten the margin of the SVM's hyperplane.

With fine histogram binning and other non-blind data-specific parameter adjustments, Dalal and Triggs were able to achieve up to 89% detection rates. By their estimate, about 10% of that success is a result of those adjustments. They performed their experiments by introducing the new "INRIA" database of 1805 images of humans (scaled, generally downscaled, to 64×128) and the original stills from which these were cropped. It is worth noting that the pedestrians are generally the focus of each photo, though in various poses and backgrounds.

The result of their algorithm on our daytime data was a recall rate of 15.3%, and a false positive rate of 85%. The algorithm's low pedestrian detection rate in our daylight test sequence was somewhat surprising. The Dalal and Triggs scoring system is designed for detection of *individual* people, so its detection score on our database is somewhat artificially deflated. An obvious difference between the INRIA and CamVid Database annotations is that our bounding boxes sometimes enclose multiple people if they overlap in the image. This difference between the class labels in the CamVid Database and object labels in the "INRIA" set hurts both the recall and false-positive scores.

Further, the CamVid Database has pedestrians with varying degrees of fractured occlusions, so the high false-positive rate is, in part, due to individual pedestrians being detected as multiple people stacked on top of one another. The algorithm is multi-scale, so seems to handle many cases where the person is approximately 64×128 pixels and larger, but frequently fails when the person is smaller (see Fig. 10). Overall, the successfully detected pedestrians in the CamVid Database are an encouraging starting point which could serve to seed a tracking algorithm, which in turn, will yield training data of people appearing at much smaller (and realistic) scales.

5.3. Object label propagation

In (Fauqueur et al., 2007), we proposed a new approach for Object Label Propagation in videos. Given a single frame of labeled objects (see Fig. 5), this method propagates automatically those labels through the subsequent frames. It relies on a joint tracking of keypoints and regions in order to achieve a robust and dense pixel propagation. In the context of manual object labeling in videos, it aims at minimizing the effort of the person labeling objects for the rest of the video sequence. Alternatively, it can be used to propagate the labels produced by an Object Recognition algorithm which could then be run at a lower frame rate.

This algorithm was tested on the `0016E5_15Hz` sequence (referred to as the `CamSeq01` dataset in the paper). The algorithm was tested using the first labeled frame of the `0016E5_15Hz` sequence as an input. The labeled object regions were propagated in the next 100 subsequent frames. The propagation accuracy was measured as the rate of correctly classified pixels in the sequence. As shown in Fig. 11, the overall accuracy (i.e. averaged over all classes) decays from 97% to 53% in the last frame.

In this context, the challenge of the automatic propagation problem is to keep track of multiple objects with independent motions, without any supervision after the first input frame. Pedestrians were among the most difficult objects to track because of the lack of contrast between their dark clothes and the dark background. As a result very few keypoints were detected on them and automatic region segmentation could not always accurately

[3] For interpretation of color in Fig. 9, the reader is referred to the web version of this article.

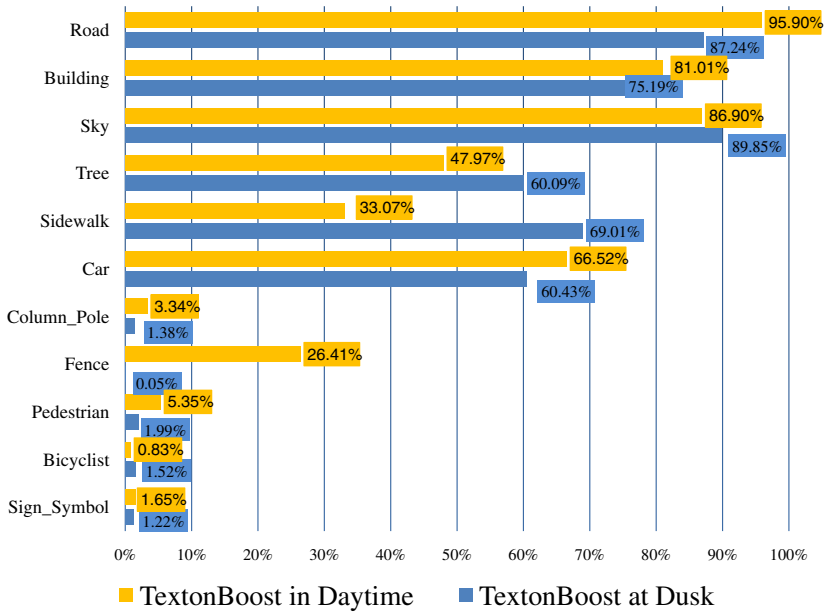

Fig. 9. Per-pixel classification rates of evaluating the TextonBoost algorithm (Shotton et al., 2006), with our modification, on the largest 11 classes in the CamVid Database. The classes are ordered here according to the number of available training pixels, *e.g.*, 4% of labeled training pixels were "Car", and the remaining classes on the list each occupied less than 1% of the labeled image area. The overall score for TextonBoost in training and testing with daytime subsets of our database is 75.02%, and for dusk it was 72.24%. Object recognition performance on the larger classes is comparable to the 72.2% reported in Shotton et al. (2006).

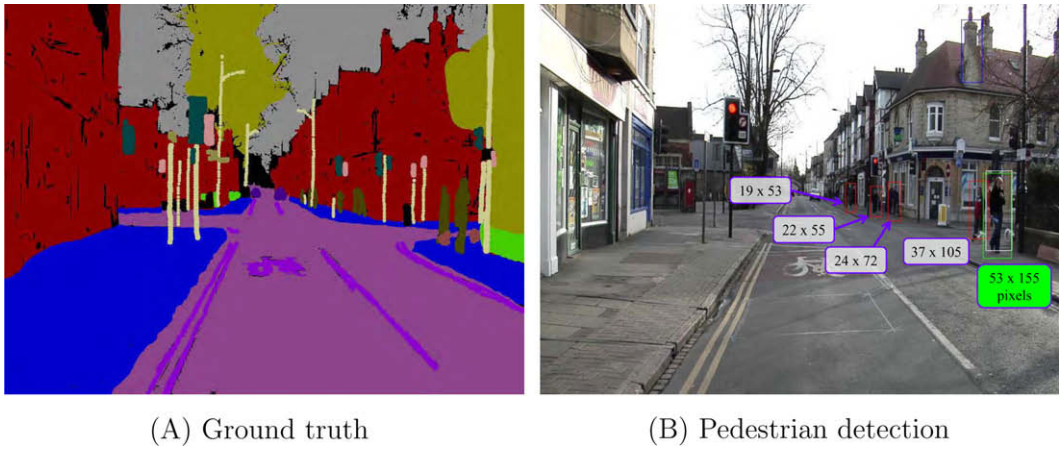

Fig. 10. (A) Human-labeled pseudocolor image showing classes visible in frame 1800 of seq05VD. (B) Example result image from testing our implementation of the Dalal and Triggs pedestrian detection algorithm (Dalal and Triggs, 2005) on our database. People who appear as size 64 × 128 pixels or larger are usually detected (shown as green boxes). People smaller than that in image-space are often not detected (shown as red rectangles). In contrast to the INRIA database, the CamVid Database has no images of people posing, so they appear here naturally as they would to a car or CCTV. (For interpretation of the references in color in this figure legend, the reader is referred to the web version of this article.)

detect them. More generally, the accuracy tended to increase with the object size and their contrast. We invite the reader to refer to Fauqueur et al. (2007) for more details and results on this approach.

6. Discussion

The long term goals of object analysis research require that objects, even in motion, are identifiable when observed in the real world. To thoroughly evaluate and improve these object recognition algorithms, this paper proposes the CamVid annotated database. Building of this database is a direct response to the formidable challenge of providing video data with *detailed semantic segmentation*.

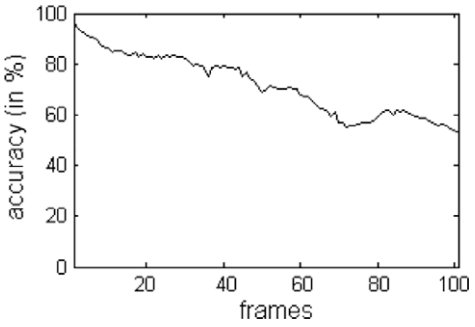

Fig. 11. Overall accuracy for label propagation averaged over all classes, tested on the `0016E5_15 Hz` sequence. The 101 evaluation frames at 15 Hz represent 6.7 s of real time.

The CamVid Database offers four contributions that are relevant to object analysis researchers. First, the *per-pixel* class labels for at least 700 images, at 1 or 15 Hz, provide the first available ground-truth for multi-class object recognition in video. The reliability of the labels is improved by requiring that two humans agreed on each frame. Second, the high-quality and large resolution video images filmed at 30 fps supply valuable extended duration footage to those interested in driving scenarios or ego-motion in general.

Third, the controlled conditions under which filming occurred allow us to present camera calibration and 3D pose tracking data for each frame in the sequences. Ideally, algorithms would not need this information, but the sub-task of auto-calibration should not stand in the way of higher level object analysis. Finally, the database is provided along with custom-made software for assisting users who wish to paint precise class-labels for other images and videos.

Through the three applications evaluated in Section 5, we have demonstrated that such a video database is a unique supplement to existing training and test datasets. It is our hope that other researchers who also work on object recognition, pedestrian detection, or object segmentation and tracking will find this database useful for evaluating their own progress. As future work, we intend to explore the efficacy of a multi-class object recognition algorithm that leverages ego-motion. Further, algorithms such as Efros et al. (2003) and Dalal et al. (2006) may be extendable to incorporate more temporal information than just frame-to-frame optical flow, eventually leading to measurable improvements in robust detection and classification.

Acknowledgements

This work has been carried out with the support of Toyota Motor Europe. We are grateful to John Winn for help during filming.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.patrec.2008.04.005.

References

Agarwala, A., Hertzmann, A., Salesin, D.H., Seitz, S.M., 2004. Keyframe-based tracking for rotoscoping and animation. ACM Trans. Graphics 23 (3), 584–591.

Bileschi, S., 2006. CBCL Streetscenes: towards scene understanding in still images, Tech. Rep. MIT-CBCL-TR-2006, Massachusetts Institute of Technology. <http://cbcl.mit.edu/software-datasets>.

Bouguet, J.-Y., 2004. Camera Calibration Toolbox for MATLAB. <http://www.vision.caltech.edu/bouguetj/calib_doc>.

Boujou, 2007. 2d3 Ltd. <http://www.2d3.com>.

Burt, P., Hong, T., Rosenfeld, A., 1981. Segmentation and estimation of image region properties through cooperative hierarchical computation. IEEE Syst. Man Cybern. (SMC) 11 (12), 802–809.

Comaniciu, D., Meer, P., 1997. Robust analysis of feature spaces: Color image segmentation. IEEE Conf. Comput. Vision Pattern Recognition (CVPR), Puerto Rico, 750–755.

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. In: IEEE Comput. Vision Pattern Recognition (CVPR).

Dalal, N., Triggs, B., Schmid, C., 2006. Human detection using oriented histograms of flow and appearance. In: Eur. Conf. Computer Vision (ECCV).

Deng, Y., Manjunath, B.S., 2001. Unsupervised segmentation of color-texture regions in images and video. IEEE Trans. Pattern Anal. Machine Intell. (PAMI), 800–810.

Efros, A.A., Berg, A.C., Mori, G., Malik, J., 2003. Recognizing action at a distance. In: IEEE Internat. Conf. Comput. Vision, Nice, France, pp. 726–733.

Facebook homepage, 2007. <http://www.facebook.com/>.

Fauqueur, J., Brostow, G., Cipolla, R., 2007. Assisted video object labeling by joint tracking of regions and keypoints. In: Interactive Comput. Vision Workshop (ICV) held with IEEE ICCV.

Fei-Fei, L., Fergus, R., Perona, P., 2006. One-shot learning of object categories. IEEE Trans. Pattern Anal. Machine Intell. (PAMI), 594–611.

Felzenszwalb, P., Huttenlocher, D., 2004. Efficient graph-based image segmentation. Internat. J. Comput. Vision (IJCV) 59 (2), 167–181.

Griffin, G., Holub, A., Perona, P., 2007. Caltech-256 object category dataset, Tech. Rep. 7694, California Institute of Technology. <http://authors.library.caltech.edu/7694>.

Hoiem, D., Efros, A.A., Hebert, M., 2006. Putting objects in perspective. In: Proc. IEEE Comput. Vision Pattern Recognition (CVPR), vol. 2, pp. 2137–2144.

Lazebnik, S., Schmid, C., Ponce, J., 2006. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: IEEE Comput. Vision Pattern Recognition, vol. 2, pp. 2169–2178.

Leibe, B., Cornelis, N., Cornelis, K., Van Gool, L., 2007. Dynamic 3d scene analysis from a moving vehicle. In: IEEE Comput. Vision Pattern Recognition (CVPR), pp. 1–8.

Leung, T.K., Malik, J., 2001. Representing and recognizing the visual appearance of materials using three-dimensional textons. Internat. J. Comput. Vision (IJCV) 43 (1), 29–44.

Müller, H., Marchand-Maillet, S., Pun, T., 2002. The truth about Corel – evaluation in image retrieval. In: Proc. Challenge of Image and Video Retrieval (CIVR2002).

Marcotegui, B., Zanoguera, F., Correia, P., Rosa, R., Marques, F., Mech, R., Wollborn, M., 1999. A video object generation tool allowing friendly user interaction. In: IEEE Internat. Conf. Image Processing (ICIP).

Martin, D., Fowlkes, C., Tal, D., Malik, J., 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. Eighth IEEE Internat. Conf. Computer Vision (ICCV), vol. 2, pp. 416–423.

Oliva, A., Torralba, A., 2001. Modeling the shape of the scene: a holistic representation of the spatial envelope. Internat. J. Comput. Vision 42 (3), 145–175.

PASCAL visual object classes challenge (VOC). <http://www.pascal-network.org/challenges/VOC/>.

Patras, I., Hendriks, E., Lagendijk, R., 2003. Semi-automatic object-based video segmentation with labeling of color segments. Signal Process.: Image Comm. 18 (1), 51–65.

Piroddi, R., Vlachos, T., 2002. Multiple-feature spatiotemporal segmentation of moving sequences using a rule-based approach. In: BMVC.

Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S., in press. Objects in context. In: IEEE Internat. Conf. on Computer Vision (ICCV).

Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T., 2005. LabelMe: a database and web-based tool for image annotation. In: MIT AI Lab Memo AIM-2005-025.

Shotton, J., Winn, J., Rother, C., Criminisi, A., 2006. Textonboost: joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Eur. Conf. Comput. Vision (ECCV), Graz, Austria.

Smeaton, A.F., Over, P., Kraaij, W., 2006. Evaluation campaigns and TRECVid. In: MIR'06: Proc. Eighth ACM Internat. Workshop on Multimedia Information Retrieval. ACM Press, New York, NY, USA, pp. 321–330.

Snavely, N., Seitz, S.M., Szeliski, R., 2006. Photo tourism: exploring photo collections in 3d. In: SIGGRAPH Conf. Proc.. ACM Press, New York, NY, USA, pp. 835–846.

The OpenCV Library. <http://www.intel.com/technology/computing/opencv/>.

The PETS, 2007. Benchmark dataset. <http://pets2007.net/>.

Wang, J., Bhat, P., Colburn, A., Agrawala, M., Cohen, M., 2005. Interactive video cutout. ACM Trans. Graphics (SIGGRAPH), 585–594.

Yao, B., Yang, X., Zhu, S.C., 2007. Introduction to a large-scale general purpose ground truth database: methodology, annotation tool and benchmarks. In: EMMCVPR, pp. 169–183.

Yuan, J., Li, J., Zhang, B., 2007. Exploiting spatial context constraints for automatic image region annotation. In: MULTIMEDIA'07: Proc. 15th Internat. Conf. on Multimedia, ACM, pp. 595–604.