Framework for generation of synthetic ground truth data for driver assistance applications

Vladimir Haltakov[1,2], Christian Unger[1,2], and Slobodan Ilic[2]

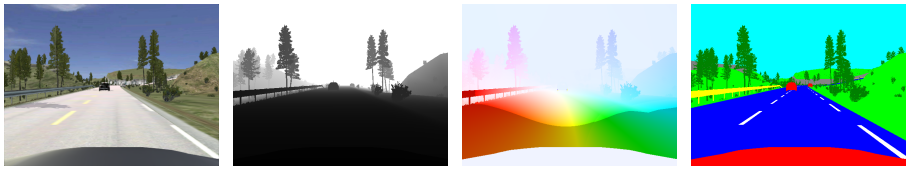
[1] BMW Group, Munich, Germany
{vladimir.haltakov,christian.unger}@bmw.de
[2] Technical University Munich, Germany
slobodan.ilic@in.tum.de

Abstract. High precision ground truth data is a very important factor for the development and evaluation of computer vision algorithms and especially for advanced driver assistance systems. Unfortunately, some types of data, like accurate optical flow and depth as well as pixel-wise semantic annotations are very difficult to obtain.
In order to address this problem, in this paper we present a new framework for the generation of high quality synthetic camera images, depth and optical flow maps and pixel-wise semantic annotations. The framework is based on a realistic driving simulator called VDrift [1], which allows us to create traffic scenarios very similar to those in real life.
We show how we can use the proposed framework to generate an extensive dataset for the task of multi-class image segmentation. We use the dataset to train a pairwise CRF model and to analyze the effects of using various combinations of features in different image modalities.

1 Introduction

The availability of high quality datasets is a crucial factor for the development of new computer vision algorithms. On one hand, learning-based methods need large amounts of ground truth data during their training phase and on the other hand, high quality data is crucial for a thorough and fair evaluation and comparison of different methods. Innovation in the computer vision and the machine learning fields has been largely driven by datasets and benchmark evaluations, which allow us to quantitatively measure the progress in the field.

Some types of data are relatively easy to obtain with high precision e.g. camera images or depth images in indoor scenarios. For such image modalities there are low-cost sensors available that can generate a lot of data very quickly. Other ground truth data, like object annotations or multi-class pixel-wise annotations of images, can be obtained by manual user annotation. Unfortunately, this is a very time consuming and expensive task. Other types of ground truth data, like depth images outdoors and optical flow, are also very difficult to obtain. Stereo cameras or laser scanners can deliver depth information, but suffer either in precision or deliver very sparse information. For optical flow there is no sensor that is able to measure it directly and manual annotation is practically impossible.



a) Camera image b) Depth image c) Optical flow d) Segmentation

Fig. 1. Example images in 3 different modalities and the corresponding pixel-wise annotations generated with the proposed framework. In the flow image, pixels with flow of more than 10 px are shown in darker colors. More examples are available in the supplementary materials.

There are already several large-scale datasets for the evaluation of stereo [18, 10], optical flow [2, 7, 26, 10] and object detection [9, 10]. However, for the task of outdoor multi-class image segmentation, there is still no extensive dataset that includes high quality video sequences of outdoor scenes, high precision dense depth maps and a large set of annotated images. Such datasets are particularly important in the field of advanced driver assistance systems (ADAS). While several datasets [4, 8, 16, 15] provide some of this data, they either have only low quality images or very sparse or imprecise depth maps (see Section 2).

One way to easily obtain large amounts of high precision ground truth data is to generate it synthetically. While it is very difficult to claim that an algorithm that performs well on a synthetic dataset will also perform well on real images, synthetic images are often used during the development of new methods in order to better understand some of their properties. For example [14] use a synthetic rotating sphere for the evaluation of scene flow and [23, 24] use synthetic traffic sequences rendered with POV-Ray to evaluate stereo and optical flow algorithms for driver assistance. Furthermore, benchmarks like the popular Middlebury dataset [2] or the recently introduced Syntel dataset [7, 26] employ synthetic images for the evaluation of optical flow and [6] use synthetic scenes for the evaluation of background subtraction algorithms.

In this work we present a new open-source framework for generating synthetic data in multiple image modalities with corresponding pixel-wise semantic annotations with focus on driver assistance applications (see Fig. 1). The framework is based on the open-source driving simulator VDrift [1], which provides a very realistic rendering engine and physics simulation. Having full access to the source code and the 3D models of the simulator, we are able to modify it, so that we can generate not only camera images, but also high precision depth and optical flow maps, pixel-wise semantic annotations and to record the exact camera pose for every frame. The proposed framework also allows the user to define specific traffic scenarios that can be of particular interest for driver assistance applications. The framework is available at `http://campar.in.tum.de/Main/VladimirHaltakov`.

Using the proposed framework we show how to generate images in 3 different image modalities and train a pairwise conditional random field (CRF) model

Table 1. Summary of the most important properties of 12 publically available multi-class segmentation datasets and our proposal.

Dataset	Image resolution	Video sequences	Depth data	Traffic scenes	Annotated images
Sowerby [12]	96 × 64	-	-	yes	104
Corel [12]	180 × 120	-	-	-	100
MSRC-21 [19]	320 × 213	-	-	-	591
Stanford Background [11]	320 × 240	-	only 3 classes	partially	715
CamVid [4], [5]	960 × 720	yes	motion stereo	yes	700
Dynamic scenes [25]	752 × 480	yes	-	yes	221
Leuven [15]	316 × 256	yes	partially	yes	70
City [8]	640 × 480	yes	stereo	yes	95
NYU Depth V1 [20]	640 × 480	yes	Kinect	-	2347
NYU Depth V2 [21]	640 × 480	yes	Kinect	-	1449
CMU RGB-D [16][1]	600 × 402	-	laser	yes	372
KITTI [10]	1384 × 1032	yes	stereo, laser	yes	-
Our framework	**2560 × 1600**	**yes**	**3D models**	**yes**	**7905**[2]

for multi-class image segmentation, similar to the one of [19]. We show how the multiple modalities allow us to explore the effects of different feature types.

2 Related Work

Since we are particularly interested in multi-class image segmentation of outdoor scenes (and especially in the context of ADAS), in this section we give an overview of some of the more important segmentation datasets. Table 1 gives a brief overview of the existing datasets and we discuss their advantages and disadvantages in detail below.

Most of the multi-class segmentation datasets, like *Sowerby* [12], *Corel* [12], *MSRC-21* [19] and the *Stanford Background dataset* [11], consist only of separate camera images with their corresponding semantic annotations. Therefore, they can only be used in segmentation methods that rely solely on texture information. Furthermore, all of the datasets mentioned above provide only relatively low resolution images (less than 320 × 240 pixels).

Some of the newer datasets, like *CamVid* [4, 5] and the *Dynamic Scenes Dataset* of [25], include video sequences which can be used to compute optical flow or structure from motion point clouds. However, these methods do not deliver high quality depth information. This could be a problem for the evaluation, because it is difficult to tell which errors are caused by the input data and which by the model itself. Both datasets do not provide any other ground truth data apart from the pixel-wise annotations.

[1] Although the *CMU RGB-D* dataset contains images of resolution 3872 × 2592, the semantic annotations are created with a resolution of 600 × 402.

[2] More images can be easily generated with the provided framework.

The *Leuven* [15] dataset and the *City* [8] dataset provide not only video sequences of road scenes and semantic annotations but also stereo camera images. While the quality of the depth maps computed with stereo matching algorithms is in general better than with structure from motion methods, the depth data can still contain many errors around object boundaries or in the presence of reflections. Furthermore, the images provided by both datasets have a relatively small resolution and only less than 100 frames are annotated.

In order to deal with the problem of providing high quality 3D data, several works introduce datasets that use an additional high precision depth sensor along with the camera: the *NYU Depth V1* [20], the *NYU Depth V2* [21] and the *CMU Driving RGB-D* [16] datasets. The first two datasets use the Microsoft Kinect to record synchronized camera and range images in indoor environments. Additionally, many of the frames have pixel-wise annotations. However, outdoor scenarios pose significantly different challenges than indoor scenes and therefore those datasets cannot be used to develop or evaluate methods that are required to work in outdoor environments, like in the case of driver assistance systems. The *CMU RGB-D* driving dataset of [16] employs a laser scanner instead of the Kinect, which allows recording of outdoor scenes. However, the images provided in the dataset are recorded at a relatively low frame rate, which makes the computation of optical flow or the usage of temporal information almost impossible. Furthermore, the used laser scanner operates in push broom mode and therefore the depth maps are sparse and do not always overlap with the camera images.

The recent dataset *KITTI* [10] is also worth mentioning even though it does not provide pixel-wise semantic annotations. The dataset consists of a large amount of stereo camera video sequences of traffic scenes synchronized with the output of a 360 degree laser scanner providing precise, but sparse ground truth depth and flow data for the lower part of the camera's field of view. The authors also provide extensive benchmarks for the evaluation of stereo, optical flow, object detection and visual odometry methods. Unfortunately, the dataset contains only bounding box annotations for cars, pedestrians and cyclists and no pixel-wise semantic labels.

3 Framework

The ground truth data generation framework is based on the open source driving simulator VDrift [1]. We considered several driving games and simulators and we chose VDrift because it provides very realistic images, a sophisticated simulation engine and a lot of different track and car models. In addition, we have full access to the source code. As we show below, the last point is essential for the generation of some of the image modalities.

3.1 Image Modalities

Since our main goal is to generate data for different image modalities, we adapted the rendering pipeline of the simulator to suit our needs. We are able to control

different rendering settings like lighting and shadow generation and we can access the 3D structure of the scenes. We are also able to control the camera pose relative to the car and in this way to simulate cameras mounted at different positions and orientations. We developed a set of OpenGL shaders that generate different image types efficiently and we are able to run the simulation in real time. The rest of the section describes in detail each of the image modalities that can be generated by our framework. One example frame can be seen on Fig. 1 and video sequences are provided in the supplementary material.

Camera images For the camera images we use the default rendering pipeline of the simulator, which is based on OpenGL. It employs several rendering techniques like anisotropic filtering, anti-aliasing, motion blur, ambient occlusion, shadows and reflections. The textures of the 3D models in the simulator are also of relatively high quality. This results in very realistic camera images with resolution of up to 2560×1600 pixels.

Depth maps From the 3D models of the scene, we are easily able to generate depth images by directly accessing the z coordinate (in the camera frame) of each pixel. Since our implementation is based on an OpenGL shader, we encode the depth information into all 3 color channels of the output image, which lieads to a precision of 24 bits per pixel. If the maximum distance is set to 1000 meters the resolution of the ground truth disparity map is 0.06 mm.

Optical flow maps The generation of ground truth optical flow is more challenging than that of the depth maps, since the required information is not directly available. In OpenGL there are two matrices that have an effect on where a 3D point should be rendered in the image - the model matrix and the projection matrix. While the camera is moving around the scene, the model matrix of each object is updated in order to account for the camera motion, while the projection matrix is usually fixed.

At each frame we provide the optical flow shader with the current model matrix for each object and the model matrix from the previous frame. With this information we can compute the 3D movement vector of each point from the previous frame to the current one. By projecting this 3D vector onto the image plane of the camera, we get the corresponding optical flow for each 3D vertex, but we still have to compute the flow value for the points of each triangle defined by 3 vertices. Interpolating the flow between the vertices in 2D would lead to wrong flow values for the triangle's points. Instead, we interpolate the 3D position of each triangle point in the current and in the previous frame and then compute the correct flow value for each image point.

As for the depth maps, we encode the flow into the 3 color channels of the output image with a precision of 24 bits per pixel. The flow values in x and y direction are encoded into separate images, from which the original flow can be easily reconstructed. This means that in the case of sequences with a maximum flow value of 100 pixels, the resolutions of the ground truth data is 0.6×10^{-5} pixels.

Pixel-wise annotations In the case of pixel-wise semantic annotations, each of the pixels in the image should be assigned a class from a set of predefined classes L that we are interested in. Here we define the set L that consists of 7 classes: SKY, TREE, GRASS, ROAD, MARKINGS, BUILDING and CAR. Unfortunately, VDrift does not support different semantic information for all of the object types listed above. For example, in the 3D model of the track, trees and buildings are indistinguishable, because they are simply modeled as 3D meshes. The road markings do not even have separate 3D structure at all. Therefore, the best way to distinguish the different object types is by their texture. We assigned a unique RGB color value to each of the classes above and modified the textures of several tracks by painting them uniformly with the corresponding colors. Rendering the scene with those textures results in a scene where each object is painted in the color specifying its class. In this case, it is very important to switch off all visual effects like lighting, shadows, reflections, anti-aliasing and mip-mapping in order to get the right color value for each pixel.

3.2 Scenario Generation

Since VDrift was originally created as a racing simulator, we modified the game engine in several ways in order to allow for the generation of scenarios that simulate real traffic conditions. Using the default AI settings, the cars would drive as fast as possible around the track. Therefore, we modified the replay system of the simulator so that we are able to manually record the movement of several cars separately and then combine all the replays into one. In this way we can create arbitrary scenarios of multiple cars driving like in normal traffic.

We also added the possibility to put stationary vehicles at arbitrary positions on the map. In this way we can simulate parked cars for more realistic scenes.

4 Evaluation

The ground truth data that we can generate using our framework gives us the possibility to explore the effects of using different image modalities for multi-class image segmentation based on a CRF. This is not possible with any of the existing datasets because they either do not include all 3 image modalities, the quality of the data is relatively poor or the amount of the training data is limited. For most of the existing segmentation datasets all of the above are true.

For the experiments we created 25 sequences on 5 different tracks simulating a car driving on country and city roads. Of those sequences, 12 were used for training and 13 for testing. The virtual camera was mounted behind the windshield of the car like a typical camera used for driver assistance systems in current vehicles. For each frame we generated images in all 3 possible modalities: camera images, depth maps and optical flow maps and the ground truth pixel-wise semantic annotations. Running the simulation at 30 frames per second we generated 7905 images. However, since at this frame rate consecutive images are very similar, we took only each 5th image, which results in a training dataset of 669 images and an evaluation dataset of 912 images.

Table 2. Pixel-wise accuracies from the evaluation of the CRF segmentation model trained on 7 different combinations of features.

Model	Accuracy	Average	Road	Marking	Building	Grass	Tree	Car	Sky
T	89.0%	70.5%	79.5%	**20.0**%	61.9%	79.2%	73.6%	88.4%	**90.9**%
D	80.6%	58.8%	88.4%	0.0%	54.2%	68.0%	25.8%	92.3%	83.1%
F	75.0%	49.3%	82.3%	0.0%	52.8%	19.4%	16.7%	91.7%	82.5%
T+D	**91.2**%	**72.2**%	88.1%	10.0%	72.5%	82.0%	70.3%	92.5%	90.2%
D+F	79.8%	57.9%	84.5%	0.0%	55.3%	73.5%	15.1%	92.1%	84.6%
T+F	90.1%	70.9%	84.7%	10.4%	64.0%	79.4%	**74.3**%	93.1%	90.6%
T+D+F	91.0%	71.2%	**88.8**%	3.2%	**71.3**%	**82.7**%	68.2%	**93.2**%	90.9%

4.1 CRF Model

For the multi-class image segmentation task we use a pairwise CRF model very similar to the TextonBoost model of [19]. Each pixel of the image is first classified with a JointBoost classifier [22] based on various features extracted from the image and then inference is performed using the α-expansion algorithm [3] on the pairwise graph. Here, we do not use the color and location unary potentials of [19], but only the classifier based unary term, while the pairwise potentials have the form of the contrast sensitive Potts model as in [19]. We also use different features than the textons of [19] in order to incorporate the texture, depth and flow information and to evaluate the contribution of the different image modalities to the performance of the CRF model.

For the **texture features** we transform the image into *Lab* color space and for each color channel we compute the mean and the variance of the first 16 coefficients of the 2D Walsh-Hadamard transform [13] at several scales around each pixel as in [25]. For our experiments we used windows of 8, 16 and 32 pixels to compute the texture features. The **depth features** consist of the 3D coordinates of the corresponding 3D point, the coordinates of the surface normal at this point and the Fast Point Feature Histogram (FPFH) of [17]. For the **flow features**, we use the 2D vector of the flow directly. In order to incorporate context information, we also include the 2D coordinates of the pixel.

The segmentation is performed on cells of 8×8 pixels instead on each pixel, but the evaluation is done at the pixel level.

4.2 Results

We analyze the importance of different image modalities for the segmentation performance. We train and evaluate 7 variants of the CRF model described above by giving it access only to the features of different subsets of image modalities: texture (**T**), depth (**D**), flow (**F**), texture and depth (**T+D**), depth and flow

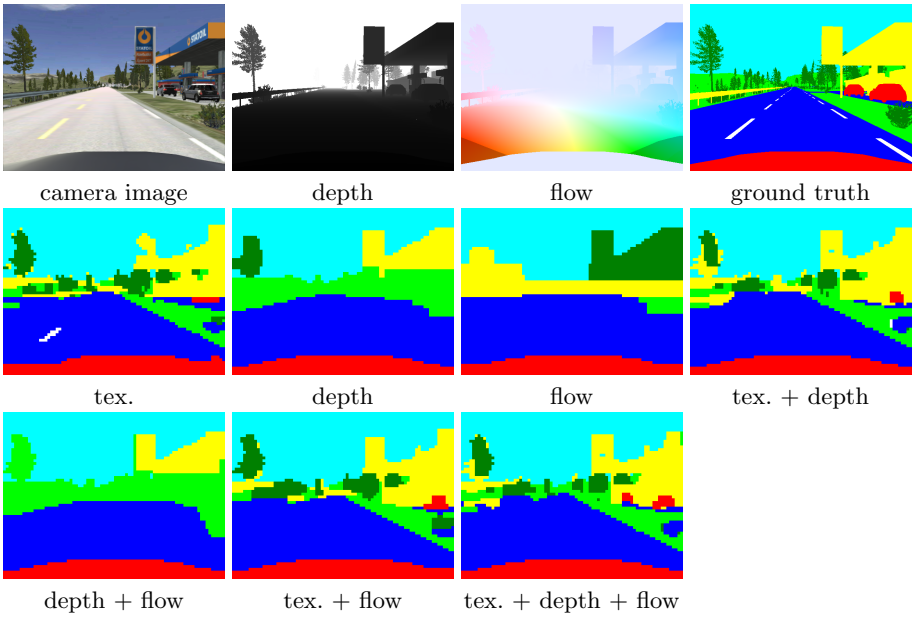


Fig. 2. Input images in 3 different modalities, the ground truth annotation and the output segmentation of all 7 models for one frame from our evaluation dataset. More result images are provided as a video in the supplementary material.

(**D+F**), texture and flow (**T+F**) and all three together (**T+D+F**). The location features are used in every configuration. In Table 2 we report the percentage of correctly classified pixels over all evaluation images, for each class individually and the average precision over all classes. In Fig. 2 we show the segmentation computed by all 7 models on one frame of the evaluation dataset. More segmentation results are provided in the supplementary material.

Using only the texture features (**T**), the CRF model is already able to achieve very good results - 89.0% overall accuracy. However, for classes with big variations in appearance, like BUILDING or CAR (compared to the other configurations), the performance is not that good, which can also be seen on the result images. Using the depth (**D**) or the flow features (**F**) alone leads to relatively poor results - 80.6% and 75.0% respectively. While the numbers may not seem too bad, we can see from the resulting images that the segmentation is much worse than in case of using only the texture features. Traffic scenes often have similar structure - sky in the upper part of the image, road in the lower part and combination of buildings, trees and cars in the middle part. This allows the CRF model to achieve high accuracy rates relatively easy relying on the location features, but in order to evaluate the real quality of the segmentation, one should look into the details. For example, we can see both from the quantitative analysis and from the result images, that when using only depth or flow, the model has difficulties distinguishing between flat surfaces like ROAD and GRASS

and the road markings cannot be detected at all. The average performance of those models over all classes is therefore also low.

It is not surprising to see that combining the texture features with either depth (**T+D**) or flow (**T+F**) features results in increased overall segmentation accuracy - 91.2% and 90.1% respectively. In most of the classes we see significant improvements, especially for ROAD, BUILDING and CAR. This is the case because those classes vary a lot in their appearance, but have well defined shapes, which can be better captured by the depth or flow features.

Combining all 3 feature types (**T+D+F**) gives overall accuracy of 91.0% that is only slightly worse (by 0.2%) than when using texture and depth features. This suggests that the information encoded by the depth and flow features is relatively similar. This is also confirmed by the model that uses them together (**D+F**). When combining those two modalities, no improvement is observed. However, in the case of more dynamic scenes this could change.

In conclusion we can say that it makes sense to use depth or flow features along with texture features extracted from the camera images, because they can improve the recognition of objects that have relatively well defined shapes like for example cars. When using relatively simple features for the depth and flow modalities, as in our case, it is not beneficial to include all three image modalities, but this may change if more sophisticated features are used.

5 Conclusion

In this paper we presented a new framework for the generation of ground truth data, which is based on the realistic driving simulator VDrift. The framework can be used to easily generate big amounts of high quality camera images, depth and optical flow maps and pixel-wise semantic annotations. Furthermore, the framework allows for the generation of specific driving scenarios that can be of particular interest for the development of advanced driver assistance systems.

We show how we can generate a dataset for the evaluation of multi-class segmentation algorithms that contains images in 3 different modalities and how we can use it to evaluate the performance of different feature types.

The proposed framework can also be used to generate ground truth data for other applications. One could easily generate images of two cameras and evaluate stereo matching algorithms given the ground truth depth data or one could use consecutive frames and the generated flow maps to evaluate optical flow and structure from motion methods. Other applications of the framework include the development and evaluation of visual odometry, scene flow and object detection methods.

References

1. http://www.vdrift.net
2. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. In: IJCV (2011)

3. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI (2001)
4. Brostow, G.J., Fauqueur, J., Cipolla, R.: Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters (2008)
5. Brostow, G.J., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: ECCV (2008)
6. Brutzer, S., Höferlin, B., Heidemann, G.: Evaluation of background subtraction techniques for video surveillance. In: CVPR (2011)
7. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: ECCV (2012)
8. Ess, A., Mueller, T., Grabner, H., Gool, L.J.V.: Segmentation-based urban traffic scene understanding. In: BMVC (2009)
9. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV (2010)
10. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: CVPR (2012)
11. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV (2009)
12. He, X., Zemel, R., Carreira-Perpin, M.: Multiscale conditional random fields for image labeling. In: CVPR (2004)
13. Hel-Or, Y., Hel-Or, H.: Real time pattern matching using projection kernels. ICCV (2003)
14. Huguet, F., Devernay, F.: A variational method for scene flow estimation from stereo sequences. In: ICCV (2007)
15. Ladický, L., Sturgess, P., Russell, C., Sengupta, S., Bastanlar, Y., Clocksin, W., Torr, P.H.: Joint optimisation for object class segmentation and dense stereo reconstruction. In: BMVC (2010)
16. Munoz, D., Bagnell, J.A., Hebert, M.: Co-inference for multi-modal scene analysis. In: ECCV (2012)
17. Rusu, R., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: ICRA (2009)
18. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In: IJCV (2002)
19. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: ECCV (2006)
20. Silberman, N., Fergus, R.: Indoor scene segmentation using a structured light sensor. In: ICCV - Workshop on 3D Representation and Recognition (2011)
21. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012)
22. Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing visual features for multiclass and multiview object detection. In: PAMI (2007)
23. Vaudrey, T., Rabe, C., Klette, R., Milburn, J.: Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In: IVCNZ (2008)
24. Wedel, A., Brox, T., Vaudrey, T., Rabe, C., Franke, U., Cremers, D.: Stereoscopic scene flow computation for 3d motion understanding. IJCV (2011)
25. Wojek, C., Schiele, B.: A dynamic conditional random field model for joint labeling of object and scene classes. In: ECCV (2008)
26. Wulff, J., Butler, D.J., Stanley, G.B., Black, M.J.: Lessons and insights from creating a synthetic optical flow benchmark. In: ECCV Workshop on Unsolved Problems in Optical Flow and Stereo Estimation (2012)