Vision-based Offline-Online Perception Paradigm for Autonomous Driving[*]

German Ros[§†], Sebastian Ramos[§], *Student Members, IEEE*
Manuel Granados[†], Amir Bakhtiary[§‡],
David Vazquez[§], Antonio M. Lopez[§†], *Members, IEEE*
[§]Computer Vision Center, [†]Universitat Autònoma de Barcelona, Campus UAB, Barcelona (Spain).
[‡]Universitat Oberta de Catalunya, Rambla del Poblenou, 156, Barcelona (Spain).
{gros, sramosp, dvazquez, antonio}@cvc.uab.es, manuel.granados@e-campus.uab.cat , abakhtiary@uoc.edu

Abstract

Autonomous driving is a key factor for future mobility. Properly perceiving the environment of the vehicles is essential for a safe driving, which requires computing accurate geometric and semantic information in real-time. In this paper, we challenge state-of-the-art computer vision algorithms for building a perception system for autonomous driving. An inherent drawback in the computation of visual semantics is the trade-off between accuracy and computational cost. We propose to circumvent this problem by following an offline-online strategy. During the offline stage dense 3D semantic maps are created. In the online stage the current driving area is recognized in the maps via a re-localization process, which allows to retrieve the pre-computed accurate semantics and 3D geometry in real-time. Then, detecting the dynamic obstacles we obtain a rich understanding of the current scene. We evaluate quantitatively our proposal in the KITTI dataset and discuss the related open challenges for the computer vision community.

1. Introduction

Autonomous driving is considered as one of the big challenges in the automotive industry. Reliable and affordable autonomous vehicles would create a large social impact. In particular, it is expected a reduction in road fatalities, a more steady flow of traffic, a reduction of fuel consumption and noxious emissions, as well as an improvement in driver comfort and enhancement of mobility for elderly and handicapped persons. Accordingly, making autonomous our mobility systems has been targeted by governments and industry for the first part of next decade. An autonomous vehicle needs to *perceive* its environment and *acting* (driving) accordingly for safely arriving to a given destiny. This paper focuses on perception.

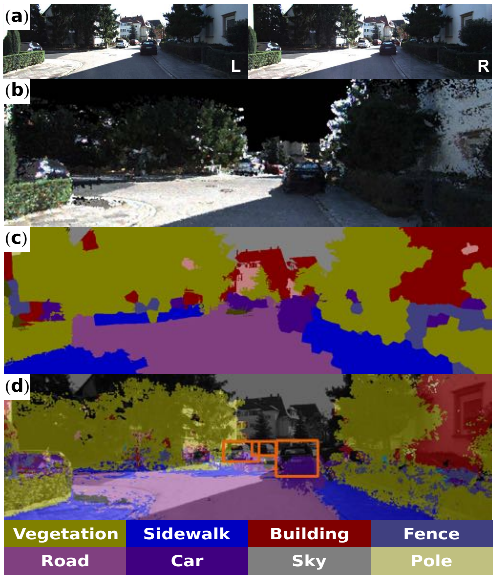

Figure 1. *Offline-Online strategy*: **(a)** urban scene that shows the input of our system, *i.e.*, stereo images. **(b)** Offline 3D scene reconstruction for maps creation. **(c)** Offline scene semantic segmentation of the 2D images to extend 3D maps with semantic information. **(d)** Online semantic and geometric information retrieval and detection of new dynamic objects.

Human drivers rely on vision for perceiving the environment. However, in recent autonomous driving attempts, as those in the DARPA Urban Challenge [6] and the demonstrators shown by Google Inc., the perception stage highly relies on an extensive infrastructure of active sensors such as lidars to create 3D representations of the environment (instantaneous 3D representations or permanent 3D maps), lidars or radars for detecting dynamic obstacles, or differential GPS to have an accurate position of the vehicle within previously generated 2D road maps. Broadly speaking, these sensors have the aim of providing a textureless 3D (*geometry*) route for planing the navigation. However, in practice, this planing also requires *semantic* information such as the type of the obstacles (*e.g.*, vehicles, pedestri-

[*] This work is supported by the Spanish MICINN projects TRA2011-29454-C03-01, TIN2011-29494-C03-02, Universitat Autònoma de Barcelona PIF program and FPI Grant BES-2012-058280.

ans) to account for their typical dynamics (*e.g.*, speed, trajectory), the state of regulatory traffic elements (*e.g.*, traffic signs, semaphores) to respect traffic rules (*e.g.*, speed limits or stopping), etc. This semantic information is extracted by the use of cameras and computer vision algorithms. Accordingly, most prototypes of autonomous vehicles incorporate several cameras, even stereo rigs for complementing the 3D information provided by active sensors.

Hence, in these approaches computer vision plays the role of a complementary technology from the very beginning. In opposition, we think that computer vision must be at the core of perception for autonomous driving, *i.e.*, as happens to be for human drivers. We do not claim that no other sensor technology will be used, however, as researchers we want to push computer vision to its limits and then analyze which other technologies can really complement it for having more reliable autonomous vehicles. Note that reducing the number of active sensors can significantly reduce the cost of autonomous vehicles, thus, maximizing the population that would have access to these products.

Beyond emulating human drivers, there are more reasons to put vision at the center of perception. Many advanced driver assistance systems (ADAS) available in the market already rely on computer vision for performing lane departure warning, intelligent headlights control, traffic sign recognition, pedestrian detection, etc. Hence, for the automotive industry camera-based systems have already moved from prototyping to serial production. Cameras are very cheap sensors (favoring affordability) with increasing resolution and easy aesthetic integration in vehicles (a *must* for selling). Moreover, their power consumption is negligible (*e.g.*, at least one order of magnitude less than active sensors like lidars), which allows a vehicle to have a 360° view coverage by using camera networks within the vehicles (*e.g.*, in the short term some vehicles will replace interior and exterior rear-view mirrors by cameras). This low power consumption is specially relevant for electric vehicles, other of the big challenges for the automotive industry.

Overall, since current autonomous vehicle prototypes include camera sensors, we aim to obtain the most from computer vision to face this challenge. In this paper we propose a vision-based framework for computing the geometry and semantics of the scene. This computation involves several modules and some of them, *e.g.* semantic segmentation, are very computationally demanding, thus unsuitable for real-time systems.

To solve this critical problem we propose a novel strategy to obtain both, semantic and geometric information in real-time via an offline-online paradigm (see Fig. 1). First, rich vision-based 3D semantic maps are created offline (see Fig. 2Top). In other words, the information that represents more computational cost is computed offline and stored. Thus, the involved algorithms do not require to sacrifice accuracy for the sake of speed. Afterwards, when a vehicle re-visits the area, the information of the map becomes available in real-time by the visual re-identification of the area and the retrieval of the semantic and geometric information that match the current vehicle viewpoint (see Fig. 2Bottom). The idea is that much of the semantic information is location dependent, meaning that the information is *anchored* to a spatial position. Therefore, by learning the semantics and by associating them to a spatial representation (map) we can recover much relevant information online. For planing the instantaneous motion of the vehicle, the information recovered from the map is extended by detecting the *new* dynamic objects (vehicles and pedestrians) in the current scene.

The proposed strategy is analogous to the way humans drive. After traversing an area several times, they have in mind a picture of the static layout, including the shape of the road, buildings, etc. The understanding of the static scene is done efficiently due to the learned model, which allows to anticipate critical areas and focusing on the localization of new dynamic objects.

Under this paradigm, the 3D semantic maps can be extended with more information, which can be used for driving safely. For instance, the location of schools, hospitals, charging points for electrical vehicles, complex roads, etc. The visual information of such maps could be frequently updated by collecting also the images that the autonomous vehicles capture for their navigation. Moreover, from the perspective of smart cities, the 3D visual information can be used not only for autonomous navigation, but also for assessing the state of the environment (*e.g.*, does the road surface has cracks or holes? are the leafs of the trees green when they should?, etc.).

Beyond the possibilities that we envisage, the first question is if our proposal is feasible, *i.e.*, whether we can retrieve the 3D semantic information attached at the current location of the vehicle in an accurate and real-time manner or not. In this paper we focus on answering this question given the state-of-the-art in computer vision for semantic segmentation [32], 3D mapping [16], self-localization [21, 26] and object detection [13]. To develop a proof-of-concept we assume vehicles equipped with a forward facing stereo rig of color sensors. This lead us to use the popular KITTI dataset [15] for quantitative evaluations. As to the best of our knowledge, this vision-based perception paradigm for autonomous driving has not been presented before. Moreover, in this context, we point out the open challenges for the core used computer vision methods.

The remainder of this paper is organized as follows. Sect. 2 presents an overview over related work. Sect. 3 focuses on the offline 3D semantic map generation and the online recovering of the semantics and 3D geometry. Reviewing open problems for this offline-online approach. Sect. 4 presents current results. Sect. 5 draws the conclusions.

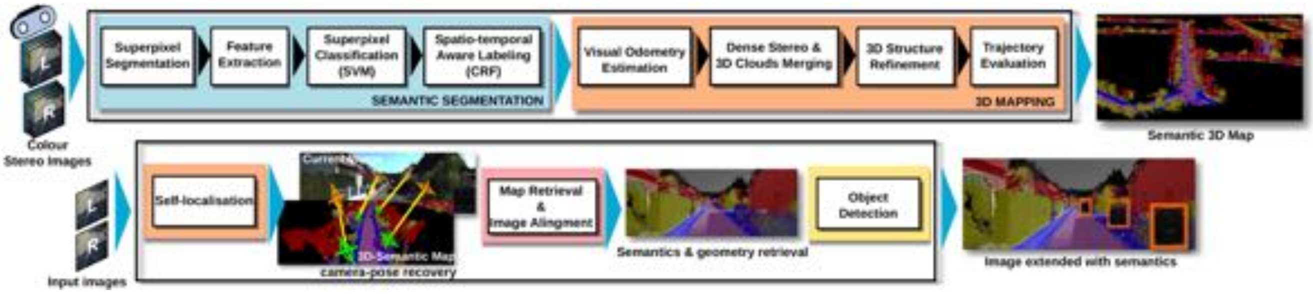

Figure 2. **Top**: *Framework for Offline 3D Semantic Map Generation*. The right part of this framework labels one of both monocular image-sequences (left or right) with a set of urban semantic classes. The right part generates a 3D map reconstruction from the stereo image-sequences and combines that with the semantic labels to produce a 3D semantic map. **Bottom**: *Framework for Online 3D Scene Understanding*. The first part of this framework recovers the vehicle pose for the input image with respect to the generated 3D map. Then, the related geometry and semantics for that pose are retrieved and aligned to the input image. Finally, dynamic objects are detected.

2. Related Work

Autonomous navigation in urban environments requires a deep and fast understanding of the scene for safe decision making regarding path planning and obstacle avoidance [2, 12, 33]. In order to address the challenge of understanding urban scenes, most research has focused in scene semantic segmentation and 3D mapping as partial solutions.

In the context of urban scene semantic segmentation, several novel approaches have been proposed [5, 31], which have provided major improvements towards a good solution for this problem. However, most of these algorithms work in the image domain, where each pixel is classified with an object label, missing important information such as general 3D structure and object geometry.

Another approach for addressing urban scene understanding is 3D mapping, which has been extensively studied [1, 30]. Nevertheless, analogous to the aforementioned case of semantic segmentation, 3D mapping approaches leave aside critical information such as object categories, which is necessary to correctly understand challenging urban environments and the interactions between their components. In fact, the popular Google self-driving project elaborates 3D maps from Lidars and then semantic information is incorporated into the 3D maps by tiresome manual annotation.

Motivated by such drawbacks of semantic segmentation and 3D mapping approaches, some recent works [28, 32] have explored the idea of combining both to create semantic 3D reconstructions of outdoor environments, which results in an advanced solution for the urban scene understanding problem. Although these approaches reach a high level of scene understanding, their use in real applications is still limited due to their extreme computational cost.

Alternative approaches [18, 24] try to alleviate the mentioned extreme computational cost of scene understanding methods, while keeping reasonable performance. Although these works showed improvements towards the mentioned objective, the reached efficiency makes them not totally suitable for challenging real-time applications.

In this sense, the approach proposed in this paper goes a step forward by factorizing the process of understanding into an offline and an online stage. Additionally, we propose an efficient mechanism to retrieve the offline precomputed information when a vehicle needs it. This formulation along with the usage of 3D semantic maps for accounting for static scene elements, lead to a promising strategy to be applied for real-time scene understanding.

3. Offline-Online Perception Approach

The vision-based understanding of urban scenes is a complex process that usually leads to very time consuming solutions, preventing from their use on-board vehicles. We overcome this limitation by using two well-defined stages, called *offline* and *online* (real-time on-board). The offline stage (see Fig. 2Top) estimates the semantic and geometric information of the static world. This rich information is then represented in a 3D semantic map as shown in Fig. 1. This map is reused by the online stage, which happens when a vehicle revisits a mapped place (see Fig. 2Bottom). There, our mechanism of semantics retrieval will bring back all the previously computed information that is relevant in our current context. This becomes possible through an accurate process of visual re-localization, that connects the offline information with the online scene. This strategy provides us with important information at a low computational cost, which must be complemented with the knowledge of dynamic objects. To this end, state-of-the-art object detection and tracking methods can be applied.

3.1. Offline 3D Semantic Map Generation

Despite the effort of the computer vision community [25], semantic segmentation keeps being computationally too expensive and often this issue is addressed at expense of loosing accuracy. The estimation of geometric information suffers from similar problems. When this information is needed to perform navigation and planning within cities, dense [16] and large reconstructions [30] are needed. Low-level navigation tasks require high accuracy, what makes necessary the use of sophisticated methods [1], again with high computational cost. Our approach circumvents these problems, from the application viewpoint, by performing both tasks offline, favoring accuracy. Color stereo sequences are fed to our offline map creation method, which first performs semantic segmentation and afterwards creates a 3D map enriched with semantic labels (Fig. 2Top).

3.1.1 Semantic Segmentation

Our semantic segmentation framework uses a Conditional Random Field (CRF) based approach that performs superpixel-wise classification on video-sequences. We consider a set of random variables $\zeta = \{\zeta_1, \ldots, \zeta_N\}$, representing superpixels. Each random variable takes a value from a set of labels $\mathcal{L}$, defined as $\mathcal{L} =\{$*building, vegetation, sky, road, fence, pole, sidewalk, sign, car, pedestrian, bicyclist*$\}$. Assigning labels to variables (superpixels) is done by a CRF, which represents the conditional distribution of ζ given a set of observations Y. In our case the observations are scores from a SVM classifier.

Let define the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ indexes the nodes that correspond to random variables, and $\mathcal{E}$ is the set of undirected edges representing compatibility relationships between random variables. Also, $\mathcal{N}_i$ is defined as the set of spatial neighbors of the random variable ζ_i. From these elements our goal is to find the optimal labeling

$$\zeta^*=\text{argmin}_\zeta\ E(\zeta)=\sum_{i\in\mathcal{V}}\psi_i(\zeta_i;\theta,Y)+\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}\psi_s(\zeta_i,\zeta_j;\theta,Y)\ \ ,$$

with θ being the parameters of the CRF model learned in a training stage. This minimization problem is solved using a graph-cut based Alpha Expansion algorithm [4]. $E(\cdot)$ is composed by unary and pairwise potentials.

Unary potentials $\psi_i(\zeta_i)$ are the scores obtained from the semantic classifiers applied at each superpixel independently after the over-segmentation of the image (done by the Mean-shift algorithm [11]). Since we perform this over-segmentation in continuous image sequences, we also consider temporal coherence. This is done by assuming that nearby superpixels and with similar colors should have similar motion, as in [9]. Then, we describe each superpixel using the multi-feature variant [20] of TextonBoost algorithm [29]. After that, we obtain the classification scores for each superpixel with linear SVMs. On the other hand, for pairwise potentials ψ_s we use the differences in color between neighboring nodes. These potentials penalize neighboring nodes with different labels, depending on the color differences, *i.e.* $\psi_s(\zeta_i, \zeta_j) = c_{ij}\mathbf{I}[\zeta_i \neq \zeta_j]$, where $\mathbf{I}[\cdot]$ is an indicator function and c_{ij} is a similarity measure of the color between the i-th and the j-th superpixels. This color similarity measure comes from the norm of the difference between the mean RGB colors of the superpixels indexed by i and j. This is, $c_{ij} = ||c_i - c_j||_2^2$, where c_i is the 3D vector of the mean of the RGB color of superpixel i.

Open challenges in semantic segmentation. Although very promising, the accuracy of current semantic segmentation approaches is far from being the required for real driving applications. There are several reasons behind these levels of accuracy, such as the lack of enough training images (with its correspondent pixel-labeled ground truth) or the lack of models that fully exploit available information like the 3D or other source of high-level information. Regarding the first problem, we have been working on the annotation of several hundreds of urban images, but due to the complexity of this task a common effort of the community would be required. The last issue is fortunately receiving more attention since new 3D-oriented CRFs have been proposed, although it seems there is much to improve still.

3.1.2 3D Mapping

We have made use of some of the advantages of offline computing to create a simple but accurate 3D mapping procedure. Our method works by decoupling the estimation of camera poses (trajectory) and 3D structure. This process is carried out in four steps: (i) visual odometry (VO) estimation; (ii) dense stereo computation and 3D point clouds merging; (iii) 3D structure refinement; (iv) trajectory evaluation (see Fig. 2Top). During this process we assume a calibrated stereo rig as our sensor; a device that major car manufacturers are already incorporating in their cars.

Firstly, we extract a sparse set of features from the four views produced by the stereo rig in previous and current time instants. These features are based on a combination of FAST-BRIEF [7, 8] methods and serve to associate features along the four views. We estimate a large number of matches N_{feat} (in the order of thousands) and then feed a VO method specialized on exploiting large amounts of constraints to estimate trajectories robustly [26]. To this end, input matches are reduced to a compact structure called Reduced Measurement Matrix (RMM) which is used to evaluate H_{cand} models very quickly. As a last step, the most promising hypotheses, H_{sel}, are combined with robust ℓ_1-averaging on the $\mathbb{SE}(3)$ manifold, which is the natural manifold of camera poses. It is worth mention that, for this application, no global coherence in the map is needed. The approach can still work well with coherence in a local basis, what makes the task of the VO algorithm simpler.

The next step consists of estimating dense stereo for each pair of input images with the local stereo algorithm presented in [16]. Next, each disparity map is converted to a point cloud with the calibration information and then they are added to the global map. It is here, where the labels from semantic segmentation are retrieved to the map, one per voxel (in case of several labels for the same voxel, the most voted would be selected). Afterwards, point clouds are combined by integrating camera poses from step (i) and projecting all the 3D structure accordingly.

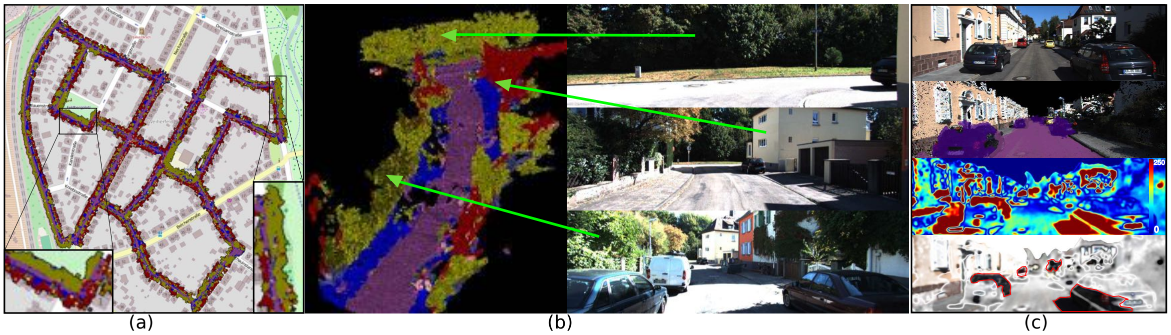

Figure 3. (a) Semantic 3D reconstruction of the sequence 00 from the KITTI [15] Visual Odometry benchmark, overlayed with the corresponding *OpenStreetMap* image. In (b) the arrows associate the different segments with the 3D model. In (c) the top image corresponds to the current frame which is related to the retrieved semantic 3D map, shown below it. The third image shows scene variations between the current scene and the original map. The final image shows how these variations can be exploited to detect possible obstacles.

To improve the resulting map, the 3D structure is first filtered out to remove spurious data. Given a set of points ξ_X in the neighborhood of X, $\xi_X = \{X' : \|X - X'\| < \epsilon\}$, when $\xi_X = \emptyset$ the point X is deleted. This simple test improved our results notably. Voxels of dynamic classes (*e.g.*, car, pedestrian, etc.) are also deleted, since they do not contribute to our purposes and would cause problems during the semantics retrieval stage. As a consequence of this process holes can appear in the map. In our experiments this did not lead to any complication as they were spurious, but we think this is an issue to address in the future.

Within this step we also re-estimate the position of all remaining points by applying a Moving Least Square method (MLS) [14]. MLS finds a polynomial p of degree d that minimizes the cost function given by $p* = \min_p \sum_{i=1}^{N} (p(X_i) - f(X_i))^2 w(\|X - X_i\|)$. Here, $w(\cdot)$ is a weight function tending to zero as their argument increases. The obtained polynomial serves to produce smoother surfaces and allow us to perform up-or-down sampling of the point cloud according to our needs. Finally, we evaluate the geometrical quality of the 3D map using *OpenStreetMap* to guarantee a minimum level of quality in local regions (see Fig. 3a). This process is currently done manually, but it could be unnecessary in the future.

***Open challenges in mapping*.** The most critical problem is how to deploy this technique at a large scale (*e.g.*, city size), since every street needs to be mapped. Projects like *OpenStreetView* have taught us that these kind of problems are solvable by having a fleet of distributed vehicles. We believe that the use of low-cost stereo sensors would make the creation of a fleet affordable for city councils and even small companies, leading to a practical solution. A closely related problem is how to maintain a map useful during long periods of time. The representation used should be rich enough to neglect the influence of small changes and should also be easy to update. Updating a map after big changes of the scene is still an open problem. It requires the detection of the changes and their inclusion into the map. The complexity of adding information to a map depends on the selected representation. For instance, when parametric models are used, modifying an area may lead to an expensive recomputation process; while, if maps are presented as set of images, updates just imply to add or change some images.

3.2. Online Retrieval of Semantics

The on-line semantics retrieval begins with the re-localization of the vehicle. We use a combination of place recognition, geometric alignment and VO. The place recognizer finds a previously visited area, simplifying the task of the geometric aligner, which recovers the vehicle pose (location and orientation) with respect to the map. Then, to reduce computational complexity, the system keeps performing re-localization by mean of VO, during N_{VO} seconds (currently set to two for a 25 fps acquisition system).

Place recognition is carried out by the algorithm described in [21]. It uses a holistic descriptor to describe each of the captured images during the map creation process, and then creates a localization database with them. Afterwards, when the method is called in the online stage, it provides a rough position within the map. This is given in terms of the most similar images to the current one (query image). The algorithm was specifically designed to be tolerant to changes of illumination and simple visual scene variations [21], which adds a fair level of robustness to the re-localization process, allowing us to perform the task even under different light or weather conditions.

After the place recognizer, Dynamic Time Warping (DTW) is performed [21, 27] to avoid ambiguities in the retrieval process. DWT uses the information of a sequence of consecutive matches to improve the results. Finally, the information provided by the holistic descriptor and the DWT is refined by using geometric information. For this, while constructing the map, we also extract multi-scale SURF [3] features and store their associated descriptors together with the 3D position of keypoints in the localization database. During the online stage, SURF 2D features are computed in

3D map creation	
VO feat. extraction & matching	0.1s
Robust camera-pose estimation	0.2s
Dense stereo computation	0.26s
Point cloud filtering & MLS	3s
Topological map creation	
Multi-scale SURF keypoints extraction	0.1s
Multi-scale SURF feat. extraction	0.2s
Dird image indexing	0.005s
Sub-total for 3D reconstruction	3.865s
Temporal semantic segmentation	
Optical flow	28.5s
Super-pixels compt.	2.3s
Feature extraction	93s
Classification	0.28s
Inference	1.85s
Superpixel tracking	20s
Labels temporal consistency	10s
TSS Sub-total	**155.93s**
Overall process per image	**159.8s**

Table 2. Times for the semantic 3D map creation

Loop-closing (LC)	
Dird image query + DTW	0.006s
Multi-scale SURF keypoints extraction (aprox. 200)	0.09s
Multi-scale SURF feat. extraction (aprox. 200)	0.07s
SURF feature matching	0.005s
Image-3DMap alignment	0.010s
Visual Odometry (VO)	
VO feat. extraction & matching	0.01s
Robust camera-pose estimation	0.015s
Sub-total for semantic and geometric inf. retrieval	*0.206s*
Object detection (in parallel @ 12 cores)	0.22s
Overall process per image (avg. case, VO)	**0.245s**
Overall process per image (worst case, LC)	**0.456s**

Table 3. Times for the online stage (retrieving & detection)

Table 1. Semantic segmentation accuracy (%) for a subset of KITTI seq 07. TSS: directly applying semantic segmentation (Sect 3.1.1), i.e., online but not real-time. SSR: real-time retrieving of the semantics from the pre-computed 3D semantic map (Sect. 3.2).

	Building	Tree	Sign	Road	Fence	Pole	Sidewalk	Overall	**Average**	**Global**
Temporal Semantic Segmentation TSS	84.3	92.8	17.1	96.8	62.9	2.1	75.2	85.6	61.6	51.2
Semantic Segmentation Retrieval SSR	79.2	84.8	12.8	94.9	49.9	2.9	81.2	81.1	58.0	47.1

the current image and we match them against the 3D points of the database. Such 3D points are those retrieved for the *locations* given by the DWT. From $3D \leftrightarrow 2D$ matches we estimate the current vehicle pose by solving a PnP problem [22], which results very efficient.

For computational reasons, we complement the previous method with the fast VO algorithm proposed by [26]. After a satisfactory localization is achieved, VO is activated and performs the localization during several frames, until the previous procedure is called back again.

Open challenges in re-localization. The main limitation of current place recognition methods is in their capability of coping with extreme visual changes due to seasonal changes and other environmental effects. This topic is gaining recognition in the literature [10, 19, 21, 23], but still is a young problem. At the present, the most solid approach seems to be to maintain a scheme of frequent updates of the desired areas, so reducing the magnitude of the changes.

4. Experiments

To validate our approach we require at least two different sequences of a common area; the first sequence is used to build the 3D semantic map offline, while the second is for recovering the semantics online. We also require these sequences to contain stereo and colour information. At this point authors would like to highlight the absence of this kind of datasets in the public domain. The closest dataset to our requirements is the KITTI benchmark suite [15] as it contains urban environment image-sequences taken from a car, what recreates the conditions of our problem and also includes two different sequences with a partial overlapping. Such sequences are the 00 and 07 from the Visual Odometry dataset, which also include stereo and color data. The overlap ranges from frame 2607 to 2858 in sequence 00 and from 29 to 310 in sequence 07. No further comparison with other sequences could be done due to the aforementioned absence of publicly available data, although we consider it enough as proof-of-concept. All the experiments were carried out in an Intel Xeon 3.2GHz with 64GB of RAM and 12 processing cores, although this parallelism is exploited only by the object detector that we include for measuring the speed of a more complete online system.

4.1. Offline 3D Semantic Mapping

The extraction of the semantic information starts by over-segmenting the images with an average of $N_{\text{sp}} = 500$ superpixels per image and then describing them by a combination of $N_{\text{ft}} = 5$ complementary features. After that, linear SVMs are fed with the calculated features in order to classify each superpixel in a one-vs-all fashion. The classification scores are combined with color similarity histograms between neighboring superpixels in a pairwise CRF to infer the most probable class label for each superpixel, as explained in Sect.3.1.1 (see Fig. 1c). In addition to the 70 ground truth images of the KITTI odometry dataset released by [32], we have manually annotated a set of 146 images more. From the set of 216 images we use 170 for training and 46 for testing. A quantitative evaluation of our semantic segmentation method is summarized in Table 1(right-row).

In order to create the 3D map we estimate the vehicle trajectory, extracting an average of $N_{\text{feat}} = 3000$ features per image and then feeding them to the visual odometry method (set up with $H_{\text{cand}} = 10000$ and $H_{\text{sel}} = 5000$). The result of dense stereo and point cloud merging was upsampled by MLS ($d = 20$), leading to a final point cloud with more than 2 billion points (stored in an efficient octree structure). For the topological layer we included about 1500 multi-scale features, along with their associate 3D information, per each of the 4541 training images[1]. The quality of the generated map is shown by Fig. 3(a,b). Thanks to *OpenStreetMap* we can see that our map is geometrically correct, and even semantics are matched (see how vegetation regions and buildings coincide).

4.2. Online Retrieval of Semantics

We measure the retrieving quality of both, semantic and geometric information, for the map created from sequence 00. Again the testing sequence is 07, due to its partial overlapping with 00. The system starts localizing itself by means of the topological layer and the loop-closure mechanism. In our experiments, the topological layer presents perfect precision for a recall of 0.64. These results, although promising, should be improved for real applications. The average pose alignment errors for the 282 images of testing remains under $20cm$ for the translation and $0.6deg$ for the rotation, allowing for an accurate alignment.

[1] Labelled images available at www.cvc.uab.es/adas/s2uad

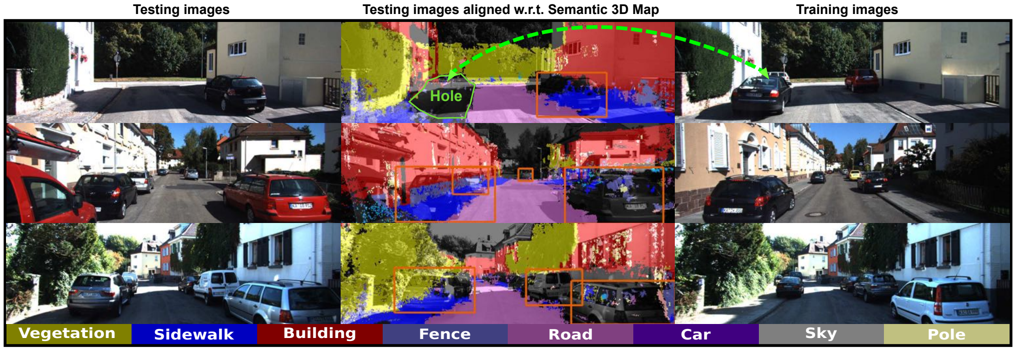

Figure 4. Semantics retrieval and object detection. The arrow highlight changes between the training and testing scenes. In particular, it is illustrated how our approach deals with cases such as objects that disappear in the testing scene with respect to the training.

As our alignment is precise, the semantic information can be correctly retrieved, matching the correct areas of the current image. To validate this in a quantitative fashion we have manually labeled 30 extra image of sequence 07, which were uniformly sampled from the area overlapping with sequence 00. The labeled images served to evaluate the difference in precision of running the Temporal Semantic Segmentation (TSS) algorithm on the images and performing the proposed semantic segmentation retrieval (SSR). To this end, we only consider the static classes and ignore the holes from the training sequences. Table 1 shows average precision per class for both methods. It can be seen that crucial classes, such as road or sidewalk are accurately retrieved, while the remaining classes are still fairly recovered. In the case of sidewalk, better results are achieved due to subtle variations between sequence 00 and 07 for the area under study, leading to a better segmentation of the training sequence and to a final higher accuracy. These results position our approach as a promising alternative to address the problem of understanding urban environments.

From a qualitative perspective, Fig. 4 illustrates the results after retrieving semantic and geometry information to the current testing images. The retrieved information is just partial in regions where the depth is not accurate enough (*e.g.*, top of buildings), although this does not compromise the approach performance. Moreover, as dynamic objects are removed from the map, holes might appear. This phenomenon is shown by Fig. 4Top, where the retrieved information has a clear hole due to the presence of a car in the images used for the reconstruction of the scene. This problem can be solved by updating the map each time the same area is traversed. Fig. 3c shows an example of the retrieved semantic and geometric information.

4.3. Timing Performance

Analyzing Tables 2 and 3 we see that our proposal is better suited for real-time performance. The cost of computing high accurate semantic segmentation ascends to $156s$ per image, while creating a 3D map that incorporates that information along with relevant geometry takes just four more seconds. As Table 2 shows, the offline extra seconds required for the 3D map creation, allows to later retrieve all semantics and geometry in real-time. This retrieving is based on the self-localization process that takes 0.21 seconds on average, when performing loop-closure. However, after the closure, visual odometry takes the control during several frames, what leads to better times and the possibility to have semantic and geometric information in real-time.

Additionally, in Table 3 we include the running time for the part-based object detectors (vehicles and pedestrians) [13], which in our current implementation is $0.22s$ per image. However, it is important to note that this time can be reduced by using state-of-the art detection approaches, which exploit stereo information [17].

4.4. Extensions

Due to the versatility of our approach, there are different extensions that can be applied to support the online scene understanding. For instance, dynamic object detection can be constrained to specific areas of the images. As proof-of-concept we use a retrieved model to compute the discrepancy with corresponding current scene by means of image analysis techniques. From this variation we can focus our attention in more specific areas to look for objects. Fig. 3c illustrates this case, where the possible positions of dynamic obstacles are highlighted to ease detection. Furthermore, due to the properties of our maps it is possible to keep adding further relevant information about the urban context, opening a door for high-level reasoning in real-time.

5. Conclusions

We have proposed a novel paradigm to address the efficiency problems of vision-based autonomous driving approaches in order to perform real-time understanding of the environment. To this end we have designed a novel offline-online strategy which has proved to be very efficient and precise. During the offline stage a 3D map of a large urban environment is created and then extended with semantic information about the fundamental elements of the scene. This map is brought to life when a previously mapped area is visited and, in consequence, all the semantic and geometrical information is retrieved to the current image. The retrieval process is accurate enough and we showed its possible use for real-time obstacle spotting. Furthermore, we have presented good qualitative results on the challenging KITTI dataset for urban environments. From a quantitative point of view, we have demonstrated the benefits of using our strategy in terms of efficacy and real-time capabilities.

As future work it is important to address several of the presented open problems, which are critical for the correct implantation of this strategy. Scalability issues require further attention in order to extend this approach to city-size maps. Furthermore, a map upload strategy would be required in order to accommodate incidental changes in "static" structures. Both, semantic segmentation and object detection methods must be improved in terms of accuracy. In the case of the former, more ground truth data is needed to achieve better results. Finally, but equally important, the self-localization needs to be improved in terms of better recall and further tolerance to severe visual changes.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building rome in a day. In *ICCV*, 2011. 4323, 4324

[2] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 2006. 4323

[3] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Speeded-up robust features (SURF). *CVIU*, 2008. 4325

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE T-PAMI*, 2001. 4324

[5] G. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, 2008. 4323

[6] M. Buehler, K. Iagnemma, and S. Singh, editors. *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*. Springer Tracts in Advanced Robotics. Springer, 2010. 4321

[7] M. Calonder, V. Lepetit, P. Fua, K. Konolige, J. Bowman, and P. Mihelich. Compact signatures for high-speed interest point description and matching. In *ICCV*, 2009. 4324

[8] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *ECCV*, 2010. 4324

[9] J. Chang, D. Wei, and J. Fisher. A video representation using temporal superpixels. In *CVPR*, 2013. 4324

[10] W. Churchill and P. Newman. Experience-based Navigation for Long-term Localisation. *IJRR*, 2013. 4326

[11] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE T-PAMI*, 2002. 4324

[12] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski. Self-supervised monocular road detection in desert terrain. In *RSS*, 2006. 4323

[13] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE T-PAMI*, 2010. 4322, 4327

[14] S. Fleishman, D. Cohen-Or, and C. Silva. Robust moving least-squares fitting with sharp features. *ACM Trans. Graph.*, 2005. 4325

[15] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012. 4322, 4325, 4326

[16] A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *ACCV*, 2010. 4322, 4323, 4324

[17] D. Gerónimo, A. Sappa, D. Ponsa, and A. López. 2D-3D-based on-board pedestrian detection system. *CVIU*, 2010. 4327

[18] H. Hu, D. Munoz, J. A. Bagnell, and M. Hebert. Efficient 3-d scene analysis from streaming data. In *ICRA*, 2013. 4323

[19] E. Johns and G.-Z. Yang. Feature co-occurrence maps: Appearance-based localisation throughout the day. In *ICRA*, 2013. 4326

[20] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Associative hierarchical random fields. *IEEE T-PAMI*, 2013. 4324

[21] H. Lategahn, J. Beck, B. Kitt, and C. Stiller. How to learn an illumination robust image feature for place recognition. In *IV*, 2013. 4322, 4325, 4326

[22] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An accurate O(n) solution to the PnP problem. *IJCV*, 2009. 4326

[23] M. Milford and G. Wyeth. SeqSLAM : visual route-based navigation for sunny summer days and stormy winter nights. In *ICRA*, 2012. 4326

[24] D. Munoz, N. Vandapel, and M. Hebert. Onboard contextual classification of 3-d point clouds with learned high-order markov random fields. In *ICRA*, 2009. 4323

[25] G. Roig, X. Boix, R. de Nijs, S. Ramos, K. Kuhnlenz, and L. V. Gool. Active map inference in crfs for efficient semantic segmentation. In *ICCV*, 2013. 4323

[26] G. Ros, J. Guerrero, A. Sappa, D. Ponsa, and A. López. Fast and robust l1-averaging-based pose estimation for driving scenarios. In *BMVC*, 2013. 4322, 4324, 4326

[27] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Trans. Acoust., Speech, Signal Processing*, 1978. 4325

[28] S. Sengupta, E. Greveson, A. Shahrokni, and P. Torr. Urban 3d semantic modelling using stereo vision. In *ICRA*, 2013. 4323

[29] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006. 4324

[30] G. Sibley, C. Mei, I. Reid, and P. Newman. Vast-scale outdoor navigation using adaptive relative bundle adjustment. *IJRR*, 2010. 4323

[31] P. Sturgess, K. Alahari, L. Ladicky, and P. Torr. Combining appearance and structure from motion features for road scene understanding. In *BMVC*, 2009. 4323

[32] J. Valentin, S. Sengupta, J. Warrell, A. Shahrokni, and P. Torr. Mesh based semantic modelling for indoor and outdoor scenes. In *CVPR*, 2013. 4322, 4323, 4326

[33] K. Wurm, R. Kummerle, C. Stachniss, and W. Burgard. Improving robot navigation in structured outdoor environments by identifying vegetation from laser data. In *IROS*, 2009. 4323