The Path Less Taken: A Fast Variational Approach for Scene Segmentation Used For Closed Loop Control

Tarlan Suleymanov[†]   Lina Maria Paz[†]   Pedro Piniés[†]   Geoff Hester[†]   Paul Newman[†]

Abstract— In this paper we propose an on-line system that discovers and drives collision-free traversable paths, using a variational approach to dense stereo vision. Our system is light weight, can be run on low cost hardware and is remarkably quick to predict the semantics. In addition to the scene's path affordance it yields a segmentation of the local scene as a composite of distinctive labels – e.g, ground, sky, obstacles and vegetation. To estimate the labels, we combine a very fast and light weight (shallow) image classifier which considers informative feature channels derived from colour images and dense depth maps estimates. Unlike other approaches, we do not use local descriptors around pixel features. Instead, we encompass label-predicted probabilities with a variational approach for image segmentation. Akin to dense depth map estimation, we obtain semantically segmented images by means of convex regularisation. We show how our system can rapidly obtain the required semantics and paths at VGA resolution. Extensive experiments on the KITTI dataset support the robustness of our system to derive collision-free local routes. An accompanied video supports the robustness of the system at live execution in an outdoor experiment.

I. INTRODUCTION

A fundamental task for a mobile robot is the ability to find and follow drivable or collision-free paths. In this paper, we propose a vision-based system that, via a variational approach, is able to segment and label semantically distinctive parts of the local scene including paths through it. Our motivation, beyond the obvious case of autonomous exploration, is the creation of a safety-net process which in the temporary absences of a localiser can still execute a safe and coherent path through its workspace. Figure 1 illustrates our approach for a single image frame.

Our system integrates different modules including dense local mapping, semantic label prediction, image segmentation, route calculation and robot control. A stereo camera is used as the primary sensing modality in this paper. Stereo cameras can provide an inexpensive and reliable means of sensing the environment for a robot at true scale if appropriately fast reliable processing schemes are deployed. Over the years, novel theoretical foundations of continuous optimisation [1], [2] and machine learning [3] for image analysis, upon which the most advanced algorithms rely, have become accessible for robotics and computer vision applications. In addition, the continuous development in parallel computing allows us to build systems that can respond in soft

[†]Mobile Robotics Group
Department of Engineering Science
University of Oxford
17 Parks Road, Oxford
OX1 3PJ, United Kingdom
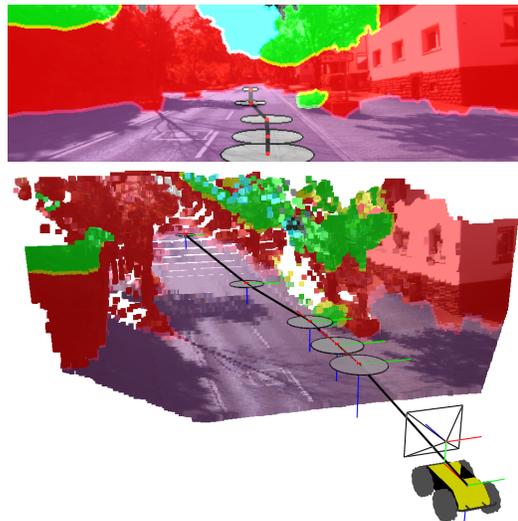tarlan,linapaz,ppinies,gjh,pnewman@robots.ox.ac.uk

Fig. 1: Our approach combines the use of a shallow classifier with convex relaxation in a multi-labelling problem to obtain semantically segmented images. Given the labels describing drivable regions, we deliver collision-free local routes to the robot controller. Top, a segmented image with the plausible path. Bottom, a dense representation of the local scene.

real time. Our algorithm for path discovery works in outdoor environments taking advantage of multiple, complementary depth and colour cues. We use these cues in a multi-label image segmentation approach. The problem is formulated in a probabilistic framework that combines machine learning with convex regularisation. In order to learn distinctive scene labels, we rely on shallow classifiers such as Random Forests (RF) [4]. This choice is driven by the intrinsic property of the RFs as low variance classifiers. As a result, they provide better generalisation by preventing from the undesirable overfitting problem. In addition, RFs explicitly allow us to model pixel-wise label probabilities with frequentist inference [5]. Moreover, RFs can easily adapt themselves to architectures supporting parallel computing and multi-threading to rapidly predict the per-pixel label probabilities. We summarise our contributions as follows:

- We demonstrate the ability of our system to run continuous optimisation at two different tasks in reasonable execution times –i.e. dense depth map estimation and multi-labeling image segmentation.
- We derive plausible routes by analysing the image semantics corresponding to drivable regions (e.g. road, ground).

We analyse the ability of RFs to combine multiple features leading to a further increase in performance when colour
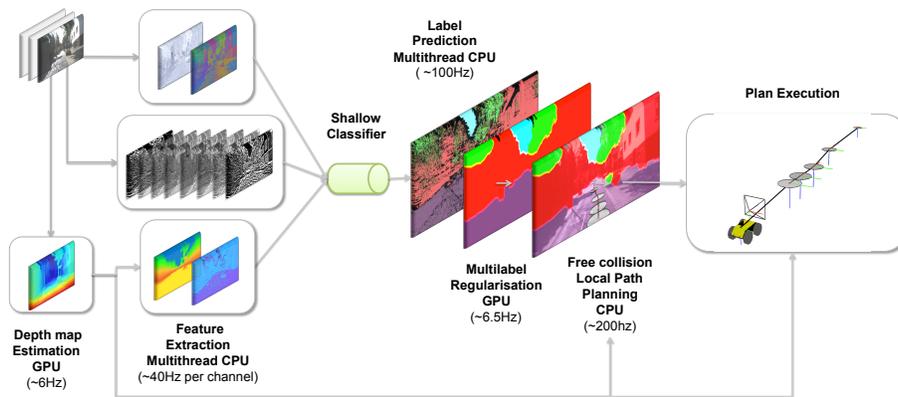
Fig. 2: Scene understanding pipeline for collision-free route following. The system considers different modules running in a main CPU multiple thread process. Stereo pairs are first processed in a dense depth map estimation task. In parallel, several CPU threads can process the rectified images to extract different colour and depth contextual feature channels. A different task is used to train our shallow Random Forest. During on-line mode, the output of the Random Forest produces the predicted per-pixel label probabilities. A new task uses this information to produced a regularised image segmentation. Given the image segmentation solution, we analyse the ground label and extract a feasible path. Finally we send the path to the robot controller.

and depth features are used simultaneously. We show how our system can rapidly obtain the required semantics – and therefore paths– at VGA resolution. Extensive experiments on the KITTI dataset support the robustness of our system to derive collision-free local routes. An accompanied video supports the robustness of the system at live execution in an outdoor experiment with a wheeled robot exploring over hundreds of metres of trajectory.

II. RELATED WORK

Over the past few years, there has been an increasing development of path-following algorithms. Many of these algorithms are not necessarily adaptive. Some rely on a priori knowledge of specific visual characteristics such as lane markers or road boundaries of the road surface [6] or their geometric structure using complementary sensor modalities such as LIDAR [7]. Other approaches employ supervised learning techniques to learn to recognise a desired class of roads by exploiting colour cues unique to the road surface in combination with segmentation algorithms [8].

In this paper, we take advantage of the two frameworks by combining dense local geometry and image colour cues. We note, however, that our ultimate goal is to find a drivable path with no assumption of any particular structure – i.e, no lane or border information is used as prior. Therefore, we deliver only collision-free paths that are suitable for pure exploration, fall-back planning (localiser failure) and off-road applications.

Related approaches have been presented in the past. For instance, in [9] a fast path following strategy for unstructured scenes is formulated as a posteriori distribution for the path given semi-dense stereo disparity, image texture and orientation features. The texture feature described is designed to adopt colour information, starting at reliably classified seed points which are provided by simple disparity based segmentation into road plane and obstacle. A different algorithm is proposed in [10] where a 1D trifocal tensor is

used to estimate parameters required for the path-following controller, through a structure from motion approach, without having to explicitly recover the 3D structure of the environment.

The focus of much of our work is the development of a path-following algorithm using scene understanding through image segmentation. Common approaches use information from dense stereo maps with Conditional Random Fields (CRFs) [11] or Convolutional Neural Nets (CNNs) [12], [13] to obtain a reasonable image segmentation – at the expense of higher computational cost to predict the per-pixel labels. A good assumption is that many scenes are a composite of vertical surfaces –e.g. buildings, vehicles, pedestrians– w.r.t the horizontal ground –e.g., road and sidewalk– with possible parts of the sky [12]. Analogously, we model the appearance of the ground using cues at pixel–level, such as colour and texture, together with contextual information from dense depth maps – in fact, they play an important role in our image segmentation task. In this work, because we require realtime performance, and in contrast to [12], [13], we use a shallow classifier rather than a deep classifier [14] to provide the data term into a down stream semantic regularisation formulated as continuous convex relaxation. Similar solutions are encountered in the literature. In [15] for instance, a set of five channels are extracted from depthmaps, however the approach does not take advantage of other colour features. Moreover, the multilabel segmentation problem is addressed as a combination of a RF classifier and a graph cut MRF based approach. In this paper, we present a solution that exploits continuous multilabel optimisation that has been shown to be superior in terms of parallelisation and runtime performance [16].

III. SYSTEM OVERVIEW

Our intermediate (but welcome) goal is to provide per-pixel semantics for the simple application of exploration with a mobile robot at near real time. To this end, we design

a system consisting of several tasks running in a multi-thread process as illustrated in Figure 2. Each left and right image of the stereo pair I_{lr}, is processed in a parallel task to estimate a dense depth map ξ. In this paper we extend the approach presented in [17] –whose solution relies on continuous energy minimisation– to estimate stereo depth maps with TGV regularisation. This choice allows us to approximate locally the ground label as an affine surface. Such approach also exploits the use of the Augmented Lagrangian (AL) method to accelerate the convergence of the primal-dual algorithm. The algorithm supports per-pixel calculations, therefore allowing us to run the task on an available GPU.

Simultaneously, several CPU threads process the left image to extract different features channels $\mathbf{Z}$ consisting of colour $\mathbf{Z}_{rgb}$, location $\mathbf{Z}_{loc}$, filter-banks $\mathbf{Z}_f$ and depth-context features $\mathbf{Z}_\xi$.

The channels are received by a different task in charge of training our shallow Random Forest. During on-line mode, the output of the Random Forest produces per-pixel u probabilities $P_{\mathfrak{T}}(u \in L_i|\mathbf{z}_u)$ of u belonging to a set of labels L_i, $i \in \{1,\ldots,K\}$) where K is the number of labels. A final task uses this information to estimate the regularised image segmentation. Analogous to the depth map estimation task, we run the regularisation on the same GPU. Given the segmentation solution, we analyse the ground label and extract a feasible path. Finally we send the path to the robot controller.

IV. PREDICTING LABELS WITH A RANDOM FOREST

A Random Forest (RF) is a popular machine learning method for classification and regression, which consists of an ensemble of decision trees $\mathfrak{T}_j$, $j \in \{1 \cdots T\}$ with predefined tree depth $d_{\mathfrak{T}}$. It has been shown that combining separate decision trees to form a forest improves performance of prediction and prevents over-fitting [4]. In our RF, each tree $\mathfrak{T}_j$ is trained individually. We follow the classical construction of the decision tree as a deterministic procedure. In order to prevent having identical trees, our trees are trained on different set of per-pixel features $\mathbf{Z}$ using a bootstrap procedure. For each tree, the same number of pixels as in the original set is randomly selected by sampling with replacement. As a result, some feature samples may appear several times, while some others could be absent. In addition, we randomly select features in each node inducing several node searching splits. Figure 3 illustrates this process.

A. Scene Features

A set of feature channels is used in order to obtain informative information about the scene by applying various transformations on the colour and depth map images. Let $\mathbf{z}_u \in \mathbf{Z}$ be a per-pixel feature vector defined as

$$\mathbf{z}_u = \begin{bmatrix} \mathbf{z}_{rgb}^T & \mathbf{z}_{loc}^T & \mathbf{z}_f^T & \mathbf{z}_\xi^T \end{bmatrix} \tag{1}$$

where $\mathbf{z}_{rgb}$ comprises an illumination invariant transform $\mathbf{z}_{ill\_inv}$ and a rg-chromaticity transform $\mathbf{z}_{rg-chroma}$ applied over the rgb pixel channels. $\mathbf{z}_\xi$ is represented by two contextual
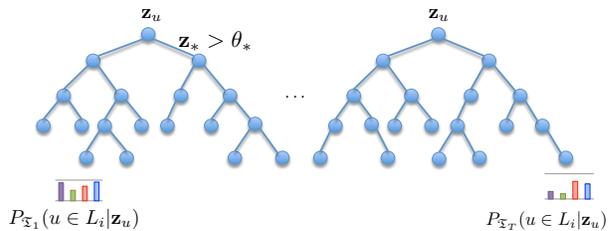

Fig. 3: The RF classifier uses per-pixel features derived from colour and depth cues. Given a new pixel sample $\mathbf{z}$, each tree classifies it in a different leaf. Each leaf saves a histogram modelling the class distribution over the samples.

transforms over the depth: first, the height of the 3D back-projection of the pixel w.r.t the ground $\mathbf{z}_{hg}$; second, the vertical disparity gradient $\mathbf{z}_{vg}$. In addition, we estimate the distance from the pixel to the horizon line $\mathbf{z}_{loc}$. Finally, we use the Leung-Malik (LM) filter bank $\mathbf{z}_f$, a collection of Gaussian and Laplacian of Gaussian filters at various scales and orientations to represent the local texture.

B. Estimation of label distribution

The probability $P_{\mathfrak{T}}(u \in L_i|\mathbf{z}_u)$ of a pixel u belonging to a particular label L_i is the result of a voting strategy. For each tree in the forest $\mathfrak{T}_j$, a subset of the components of the feature vector $\mathbf{z}_u$ are compared at each node to a given threshold θ. The comparison determines the next branch to follow until a leaf node is reached. As can be seen in Figure 3, histograms learnt during the training phase are stored at the leaves of the trees. For a given tree, the histograms contain the number of pixels per label in the training set that end up in that leaf. These histograms aim to approximate the probability $P_{\mathfrak{T}_j}(u \in L_i|\mathbf{z}_u)$. During on-line mode, the label distribution of a test pixel is given by the average of the histograms stored at the corresponding leaf of each tree in the forest:

$$P_{\mathfrak{T}}(u \in L_i|\mathbf{z}_u) = \frac{1}{T} \sum_{j=1}^{T} P_{\mathfrak{T}_j}(u \in L_i|\mathbf{z}_u) \tag{2}$$

V. REGULARISATION VIA CONVEX RELAXATION

With the initial per-pixel classification results in hand, greatly improved results can be obtained by formulating the complete image segmentation as a labelling problem with a global energy function that balances "smoothness" of the labelled segments (a prior) and per-pixel probabilities (a data term) coming from the random forest. The energy function is given by:

$$\min_{\Omega_i} \left\{ \frac{1}{2} \sum_{i=1}^{K} Per(\Omega_i) + \sum_{i=1}^{K} \int_{\Omega_i} f_i(u)du \right\} \tag{3}$$

$$s.t. \ \Omega = \bigcup_{i=1}^{K} \Omega_i, \quad \Omega_i \cap \Omega_j = \emptyset, \quad \forall i \neq j$$

where $\Omega \in \mathbb{R}^2$ represents all the pixels in the image assigned to K disjoint regions Ω_i (e.g. ground, vegetation, obstacles and sky).

In Eq.(3) the data term is given by the sum of the costs of the unary potentials $f_i(u) = -\log(P_{\mathfrak{T}}(u \in L_i|\mathbf{z}_u))$ per segmented region Ω_i. The intuition behind $f_i(u)$ is that when pixel u belongs to region Ω_i with a high probability ($P_{\mathfrak{T}}(u \in L_i|\mathbf{z}_u) \approx 1$) the cost added is negligible ($f_i(u) \approx 0$), on the contrary, low probabilities produce an increasingly unbounded cost. The main effect of the smoothness term is to reduce the perimeter of the regions $Per(\Omega_i)$ such that it tends to smooth the boundary between neighbours and delete small regions surrounded by bigger ones. In order to obtain a more convenient expression of the energy for optimisation we represent each region instead by its indicator function:

$$\phi_i(u) = \begin{cases} 1 & \text{if } u \in \Omega_i \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

The energy function in Eq.(3) can then be described by:

$$\min_{\phi_i(u)} \left\{ \frac{1}{2} \sum_{i=1}^{K} \int_\Omega |\nabla \phi_i(u)| du + \sum_{i=1}^{K} \int_\Omega \phi_i(u) f_i(u) du \right\} \quad (5)$$

$$s.t. \ \phi_i(u) \in \{0,1\}, \quad \sum_{i=1}^{K} \phi_i(u) = 1$$

where $\int_\Omega |\nabla \phi_i(u)|$ is the Total Variation (TV) of the indicator function $\phi_i(u)$ that can be shown to be equal to the perimeter of the segment.

The constraint $\phi_i(u) \in \{0,1\}$ makes the problem combinatorial and NP-hard so it can only be approximately solved. We use a known fast relaxation approach [18] that transforms the original problem into a convex one. While this relaxation is not the tightest, it produces good results in practice. The relaxation is based on allowing $\phi_i(u)$ to take values in the interval $\phi_i(u) \in [0,1]$. Since in addition we know that $\sum_{i=1}^{K} \phi_i(u) = 1$, the constraint can be relaxed to:

$$\phi_i(u) \geq 0, \ \forall i \quad (6)$$

as a result, Eq.(5) becomes a convex optimisation problem.

Unfortunately the energy cost is non-smooth due to the L1 norm that appears in the TV term. The Legendre-Fenchel Transform [19] allows us to trade the non-smoothness of the prior term for a smooth convex constrained maximisation:

$$\int_\Omega |\nabla \phi_i(u)| du = \max_{\Psi_i(u)} \int_\Omega \nabla \phi_i(u) \cdot \Psi_i(u) du \quad (7)$$

$$s.t. \ |\Psi_i(u)|_{2,1} \leq 1 \quad (8)$$

where $\Psi_i(u) : \Omega \to \mathbb{R}^2$ is known as the dual function of $\phi_i(u)$. Although this transformation seems to apparently increase the complexity, the counterpart is that we can now use well known first order methods available for smooth problems to find the global solution of the relaxed energy.

As explained below, we can easily deal with the box constraints given in Eqs.(6, 8) by projecting the solution of the optimisation at each iteration to the feasible set when it fails to meet the constraints. The equality constraint $\sum_{i=1}^{K} \phi_i(u) = 1$ can be included into the energy by introducing

Lagrange Multipliers $\Gamma(u)$. The relaxed problem to be solved is then:

$$\min_{\phi_i(u)} \max_{\Psi_i(u),\Gamma(u)} \left\{ \frac{1}{2} \sum_{i=1}^{K} \int_\Omega \nabla \phi_i(u) \cdot \Psi_i(u) du + \sum_{i=1}^{K} \int_\Omega \phi_i(u) f_i(u) du \right.$$
$$\left. + \int_\Omega \Gamma(u) \left(\sum_{i=1}^{K} \phi_i(u) - 1 \right) du \right\} \quad (9)$$

$$s.t. \ \phi_i(u) \geq 0, \quad |\Psi_i(u)|_{2,1} \leq 1$$

To solve the convex saddle point problem in Eq. (9) an iterative primal dual algorithm [1] is applied. Basically we just need to interleave gradient ascent steps for the maximisations with gradient descent steps for the minimisation at each iteration of the algorithm. In both cases we project the solution to the feasible set in case the box inequality constraints are not met. Therefore at each iteration t and per each label L_i we perform the following steps:

- Maximising $\Psi_i(u)$:

$$\tilde{\Psi}_i^{t+1} = \Psi_i^t + \sigma \nabla \bar{\phi}_i^t \qquad \text{gradient ascent}$$

$$\Psi_i^{t+1} = \frac{\tilde{\Psi}_i^{t+1}}{\max(1, |\tilde{\Psi}_i^{t+1}|_{2,1})} \quad \text{projection to feasible set}$$

- Minimising $\phi_i(u)$:

$$\tilde{\phi}_i^{t+1} = \phi_i^t - \tau(\nabla^T \Psi^{t+1} + f_i + \Gamma_i^t) \quad \text{gradient descent}$$

$$\phi_i^{t+1} = \max(0, \tilde{\phi}_i^{t+1}) \qquad \text{projection to feasible set}$$

- Maximising $\Gamma(u)$:

$$\Gamma^{t+1} = \Gamma^t + \mu \left(\sum_{i=1}^{K} \phi_i^{t+1} - 1 \right) \quad \text{gradient ascent}$$

- Over-relaxation:

$$\bar{\phi}^{t+1} = \phi^{t+1} + \theta_r(\phi^{t+1} - \phi^t)$$

where σ, τ and μ control the step size of the gradient steps. In practice, each variable is updated by performing pixel wise calculations while the gradient operator ∇ is approximated by finite differences. The parameters are set up to $1/2, 1/4$ and $1/5$ respectively through the use of preconditioning [20]. The last over-relaxation step allows faster convergence of the algorithm with $0 \leq \theta_r \leq 1$. Analogous to the depth map estimation problem, the primal dual approach followed in this section allows us to take advantage of general purpose GPU hardware for parallel computing. For a detailed derivation of the update equations, we refer the interested reader to [20].

VI. EXPERIMENTS

This section provides quantitative results for experiments carried out over two different datasets. Our heterogeneous pipeline (CPU/GPU) was tested in two different hardware architectures whose details are summarised in Table I. In addition, we show qualitative results from a live experiment carried out in an outdoor environment. Our experiments consider a parameter selection analysis as well as an assessment that guides the importance of the feature channels over the RF-based dataterm. Additionally, we make a comparison of predicted labels before and after label regularisation.

Architecture	OS	Processor	Graphics Card
Server	Ubuntu 14.01	Intel(R) Core(TM) i7 CPU @ 3.50GHz	GeForce GTX TITAN Black, 6144 MB 2880 CUDA Cores
Laptop	OSX Mavericks	Intel(R) Core(TM) i7 CPU @ 2.3GHz	Geforce 750M, 2048 MB, 384 CUDA Cores

A. Quantitative experiments

In order to evaluate the proposed image segmentation approach, we make use of a subset of the *KITTI dataset* for which ground truth is provided [21]. The dataset consists of 60 stereo pairs at resolution 1241×376 with perfect annotations for 12 semantic class labels and ground truth depth maps. For the purpose of our evaluation, we synthesise the annotations into ground, vegetation, obstacles and sky. In this case, we employ 50% of the images for training and 50% for cross-validation. An additional dataset was collected and manually labelled, *Keble College dataset*, consisting of 65 stereo frames at high resolution (1280×960). Optimised depth maps are also provided with centimetre accuracy.

1) Parameter selection for RF-based dataterm: The accuracy and computational complexity of our RF-based dataterm depends on two major parameters: the maximum tree depth and the number of trees in the forest. In order to choose the best parameters, we analyse their impact over the RF performance. For the KITTI dataset, we carry out an exhaustive search in a 2-dimensional grid representing the parameter domain. For each parameter configuration (number of trees, tree depth) we train a model. Figure 4 shows the average F1 score on the configuration space of parameters over 200 RF models. Although the performance can be optimised to find the maximum score, we found that a classifier consisting of 5 trees and 10 tree levels provides a good trade off between performance and speed.

2) Contribution of feature channels: Our assessment also takes into account the contribution of each feature channel
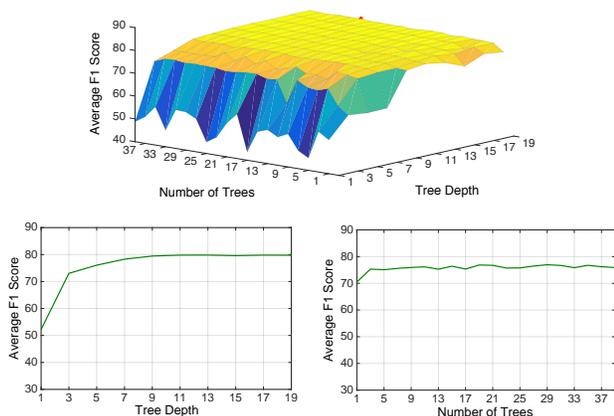
Fig. 4: Analysis of the performance of several RF-based classifiers for parameter selection. We carried out an exhaustive search in a 2-dimensional array representing the parameter space domain. For each parameter configuration (number of trees, tree depth), we train the classifier and evaluate its average F1-score. Although the performance can optimised by finding the maximum score, an a classifier consisting of 10 trees and 15 tree levels can result in a good trade off between performance and speed.
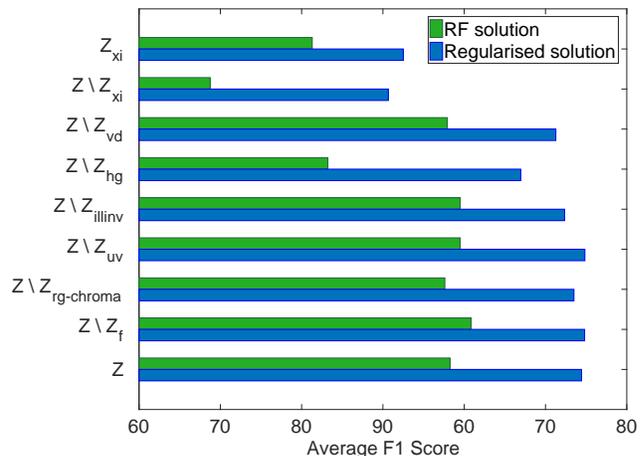
Fig. 5: Impact of the feature channels over the performance before (green) and after (blue) multi-label regularisation. In all cases, the smoothed solution outperforms the RF-classifier solution. Much worse results are obtained when depth features are left out.
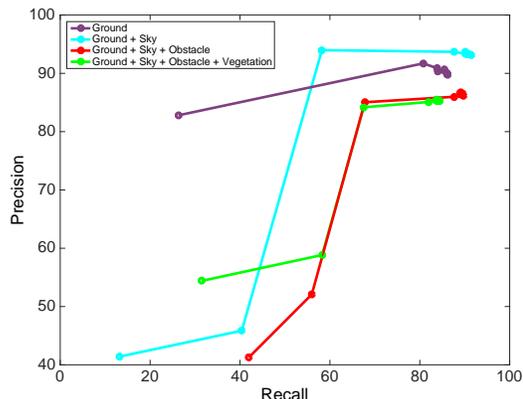
Fig. 6: Precision-Recall curves obtained by increasing the complexity of the RFs for different combination of labels. As expected, the curves exhibit better accuracy for more complex models preventing at the same time the undesirable over-fitting problem.

to the overall performance. We calculate the precision, recall and F1 score for different models. Table II details this values per label. An important part of the evaluation is to measure the impact of the depth channels (i.e, z_{ξ}). Moreover, we emphasise the importance of each depth channel by leaving one channel out at a time (e.g, $\mathbf{Z} \setminus \mathbf{z}_{hground}$). This test is also applied to the colour channels allowing us to find strong features by observing whether or not the performance drops significantly when a particular channel is missing. Figure 5 summarises this information showing the average F1 score for the KITTI dataset. Note that the overall performance significantly decreases when the depth channels are not considered. In fact, a model that uses only depth channels performs better than a model using the rest of the channels.

We can also see that texture features do not appreciably affect the performance.

Table II provides information about the accuracy achieved per label. For instance, when only depth channels are used, the ground and the sky labels already exhibit high accuracy. This is not surprising if we consider that these labels are associated to very distinctive depths. In contrast, the accuracy obtained for the obstacle and vegetation labels when just depth channels are used is lower as there is much higher variability in their depth.

Figure 6 shows how the precision and recall curves vary when we increase the complexity of the RFs for different combination of labels. As expected, the curves exhibit better generalisation (better accuracy) for increasing complex models preventing at the same time the undesirable over-fitting problem.

3) Prediction labels before and after regularisation: Table II and Figure 5 indicates the difference between the performance of the models before and after applying regularisation. Note that in all the cases, the regularised solution outperforms the RF-classifier solution.

4) Running Time: Many of the tasks involved in our pipeline can run in real time. Table III summarises the running times per task for the two testing architectures. All images are at VGA (480×640) resolution. Despite stereo depth estimation and label smoothing require more time than other tasks, the frame rates are still acceptable to provide reliable paths at live execution.

B. Live outdoor experiments

Fig. 7: Illustration of the collision-free algorithm to extract a route from the ground label.

For our live outdoor experiment we use a Clearpath Husky UGV equipped with a forward-facing PGR Bumblebee2 camera. In order to provide reliable collision-free paths, we implement an algorithm that analyses the ground label in a bottom-up direction. Figure 7 illustrates the process. The algorithm tessellates the ground label in cells with adjustable dimensions. Note that each cell contains only a sub-region of the segmented label. For each cell, we calculate the centre of mass. This simple strategy allows us to consider the shape and orientation of the drivable regions. All the points with valid centre of mass are concatenated together to form the desired path. In addition, we impose a safety margin over the robot dimensions. The robot is modelled as a circular object of 1.5 meters of diameter. Each feasible point of the circle is projected into the segmented image. We check for possible collisions if the projection intersects other labels. The path is back-projected to 3D space using the available depth map and the intrinsic camera parameters. For simplicity, our controller assumes a differential platform such that the path

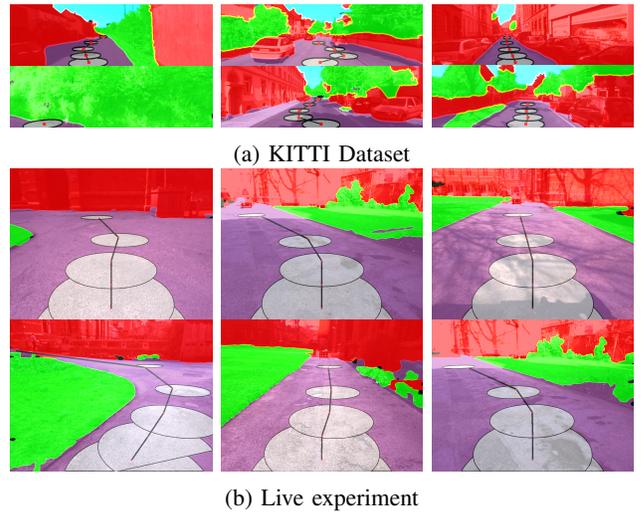(a) KITTI Dataset

(b) Live experiment

Fig. 8: Qualitative results of the path following approach. We first evaluated the collision-free approach in different scenarios. In some situations the ground segment extends in front of the robot, thus our algorithm succeeds to find a route to control the robot forwards. We show that in the presence of obstacles our approach provides collision-free paths. When the ground segment is small – for example when the robot reaches a wall– the estimated path can only provide one or no points. In this case, the robot performs pure rotation until it finds a ground region for which a plausible path can be estimated.

is executed with a constant linear velocity of 1 m/s. The angular velocity is derived from the path segments.

The collision-free path approach has been tested in different scenarios. Figure 8 shows qualitative results on the KITTI dataset and on our live experiment with a robot moving autonomously in a quad along hundreds of meters. When the ground segment extends in front of the robot and there are no obstacles, the algorithm succeeds to find the simplest route possible and plans to move the robot forwards following a straight line. When obstacles such as cars are present, our approach is able to over take the obstacles following a collision-free path. Finally, when the ground segment in front of the robot is not big enough – for example when the robot reaches a wall– no path can be provided. In this case, the robot performs a pure rotation until it finds a ground region for which a plausible path can be estimated.

VII. CONCLUSIONS

In this paper, we presented a general framework that combines a light weight (shallow) image classifier with convex regularisation for the general problem of image scene understanding. While recent approaches rely on the use of complex and deep classifiers, we demonstrate that a random forest can inform our variational formulation with very reliable label probabilities. In fact, our system requires small amounts of data during the training phase and yet produces high accurate results during testing. Finally, we showed that our system is remarkably fast to provide semantics form image data allowing a mobile robot to discover and drive collision-free traversable paths. We are also interested in comparing to alternative solutions based on deep classifiers

TABLE II: Recall, precision and F1 score for different RF models

Channels	Ground			Obstacle			Vegetation			Sky		
	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
KITTI Dataset												
$\mathbf{Z}_\xi$	91.83	87.41	89.56	69.59	65.03	67.23	57.24	63.65	60.27	98.20	98.20	98.20
	91.40	87.23	89.27	82.36	70.28	75.84	58.27	76.58	66.19	93.36	97.39	95.34
$\mathbf{Z} \setminus \mathbf{Z}_\xi$	57.29	53.25	55.20	66.49	59.72	62.93	65.98	79.15	71.97	62.89	60.02	61.42
	69.38	63.16	66.12	76.86	70.05	73.30	77.20	92.21	84.04	61.03	79.58	69.08
$\mathbf{Z} \setminus \mathbf{z}_{vg}$	89.49	89.75	89.62	83.56	72.00	77.35	66.68	79.70	72.61	93.86	98.52	96.13
	91.28	89.77	90.52	90.71	79.56	84.77	75.74	91.88	83.03	86.80	98.43	92.25
$\mathbf{Z} \setminus \mathbf{z}_{hg}$	71.03	74.24	72.60	78.63	66.36	71.97	66.16	78.60	71.85	58.40	66.86	62.35
	80.66	89.84	85.01	90.66	76.79	83.15	78.78	91.71	84.76	60.00	82.38	69.43
$\mathbf{Z} \setminus \mathbf{z}_{ill_{inv}}$	90.45	90.46	90.46	82.91	73.98	78.19	69.04	78.64	73.53	93.59	98.50	95.98
	92.61	90.78	91.68	92.89	79.73	85.81	73.99	93.10	82.45	88.53	98.49	93.24
$\mathbf{Z} \setminus \mathbf{z}_{loc}$	90.34	90.26	90.30	83.35	73.64	78.19	68.77	79.37	73.69	92.44	98.08	95.17
	92.02	91.92	91.97	91.37	82.47	86.69	79.18	91.74	85.00	88.84	98.21	93.29
$\mathbf{Z} \setminus \mathbf{z}_{rg-chroma}$	90.62	89.94	90.28	82.41	72.63	77.21	66.58	77.85	71.77	96.33	98.31	97.31
	92.71	90.99	91.84	91.18	81.78	86.22	77.35	90.98	83.62	89.06	98.26	93.44
$\mathbf{Z} \setminus \mathbf{z}_f$	90.49	91.01	90.75	82.65	75.04	78.66	70.61	78.81	74.48	97.58	98.28	97.93
	92.45	92.20	92.33	91.02	82.67	86.64	78.87	91.29	84.63	92.17	96.87	94.46
$\mathbf{Z}$	89.62	90.80	90.20	84.67	72.07	77.86	65.68	79.66	72.00	95.32	98.42	96.85
	92.01	92.32	92.17	91.13	82.24	86.46	78.60	91.34	84.49	90.61	97.32	93.85
Keble College Dataset (Full Resolution)												
$\mathbf{Z}$	99.10	94.90	96.95	89.06	93.22	91.10	78.02	98.68	87.14	—	—	—
Keble College Dataset (VGA Resolution)												
$\mathbf{Z}$	98.48	94.55	96.48	86.65	94.13	90.24	73.96	98.71	84.56	—	—	—

We compared the impact of the channels over the performance before (white rows) and after (gray rows) applying regularisation. For a better analysis, we show the independent precision-recall and F1 score per label. The first column describes the channels used for training. For instance, $\mathbf{Z}$ indicates that all channels have been used, while $\mathbf{Z} \setminus \mathbf{z}_*$ means that a particular channel has been left out. $\mathbf{z}_{ill_{inv}}$ is the illumination invariant transform, $\mathbf{z}_{rg-chroma}$ is the rg-chromaticity transform, $\mathbf{z}_\xi$ is represented by two contextual transforms over the depth. $\mathbf{z}_{hg}$ is the height of the 3D back-projection of the pixel w.r.t the ground. $\mathbf{z}_{vg}$ is the vertical disparity gradient. $\mathbf{z}_{loc}$ the distance from the pixel to the horizon line. $\mathbf{z}_f$ are the Leung-Malik (LM) filter bank $\mathbf{z}_f$.

TABLE III: Average Running time per task

Task	Server (ms (Hz))	Laptop (ms (Hz))	CPU Threads
Depth map estimation	190 ms (5.26 Hz)	1180 ms (0.85 Hz)	1
Feature Extraction	25 ms (40 Hz)	22 ms (45.45 Hz)	14
Label Probability prediction	10 ms (100 Hz)	6 ms (166 Hz)	10
Label Regularisation	155 ms (6.45 Hz)	1020 ms (0.98 Hz)	1
Route calculation	$\approx$ 5 ms (200 Hz)	$\approx$ 5 ms (200 Hz)	1

in combination with our variational approach. This will be part of our future work.

REFERENCES

[1] A. Chambolle and T. Pock, "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging," *J. Math. Imaging Vis.*, vol. 40, no. 1, pp. 120–145, May 2011.

[2] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM J. Img. Sci.*, vol. 3, no. 3, pp. 492–526, Sep. 2010.

[3] F. Schroff, A. Criminisi, and A. Zisserman, "Object class segmentation using random forests." in *BMVC*, 2008, pp. 1–10.

[4] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[5] H. Boström, "Estimating class probabilities in random forests," in *Sixth Int. Conf. on Machine Learning and Applications, 2007. ICMLA 2007.* IEEE, 2007, pp. 211–216.

[6] H. Kong, J.-Y. Audibert, and J. Ponce, "General road detection from a single image," *IEEE Trans. on Image Processing*, vol. 19, no. 8, pp. 2211–2220, 2010.

[7] P. Y. Shinzato, D. F. Wolf, and C. Stiller, "Road terrain detection: Avoiding common obstacle detection assumptions using sensor fusion," in *IEEE Proc. Intelligent Vehicles Symposium*, 2014, pp. 687–692.

[8] R. Mohan, "Deep deconvolutional networks for scene parsing," *arXiv preprint arXiv:1411.4101*, 2014.

[9] B. Hummel, S. Kammel, T. Dang, C. Duchow, and C. Stiller, "Vision-based path-planning in unstructured environments," in *IEEE Proc. Intelligent Vehicles Symposium*, 2006, pp. 176–181.

[10] D. Sabatta and R. Siegwart, "Vision-based path following using the 1d trifocal tensor," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013, pp. 3095–3102.

[11] L. Ladickỳ, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. H. Torr, "Joint optimization for object class segmentation and dense stereo reconstruction," *Int. Journal of Computer Vision*, vol. 100, no. 2, pp. 122–133, 2012.

[12] J. M. Alvarez, T. Gevers, Y. LeCun, and A. M. Lopez, "Road scene segmentation from a single image," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 376–389.

[13] A. Kendall, V. Badrinarayanan, , and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.

[14] T. Scharwachter and U. Franke, "Low-level fusion of color, texture and depth for robust road scene understanding," in *IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 599–604.

[15] C. Zhang, L. Wang, and R. Yang, "Semantic segmentation of urban scenes using dense depth maps," in *ECCV 11th European Conference on Computer Vision*, 2010.

[16] C. Nieuwenhuis, E. Toeppe, and D. Cremers, "A survey and comparison of discrete and continuous multi-label optimization approaches for the potts model," *Int. Journal of Computer Vision*, vol. 104, no. 3, pp. 223–240, 2013.

[17] P. Piniés, L. M. Paz, and P. Newman, "Dense and swift mapping with monocular vision," in *Int. Conf. on Field and Service Robotics (FSR). Toronto, ON, Canada*, 2015.

[18] C. Zach, D. Gallup, J.-M. Frahm, and M. Niethammer, "Fast global labeling for real-time stereo using multiple plane sweeps." in *VMV*, 2008, pp. 243–252.

[19] R. T. Rockafellar, *Convex Analysis*. Princeton, New Jersey: Princeton University Press, 1970.

[20] T. Pock and A. Chambolle, "Diagonal preconditioning for first order primal-dual algorithms in convex optimization," in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2011, pp. 1762–1769.

[21] L. Ladicky, J. Shi, and M. Pollefeys, "Pulling things out of perspective," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2014, pp. 89–96.