# Local Path Planning along Bush Tracks using Vision

**David Fernandez** and **Raymond A Jarvis**
Intelligent Robotics Research Centre,
Monash University, Clayton, VIC 3800, Australia
David.Fernandez@eng.monash.edu.au

## Abstract

This paper investigates the problem of autonomous mobile robot navigation in bush environments. Such environments form part of the workspace of a number of groups, such as rural search-and-rescue and bush fire-fighting crews, who could greatly benefit from helper robots. In particular, given the existence of many tracks through bush areas, the ability to autonomously drive along such tracks would help ensure robot safety and efficiency of travel, as a track by definition represents a passageway with a high likelihood of being clear. The novelty of the work presented comes from the sole use of passive vision to find obstacles and locate tracks, allowing a safe local path to be determined.

## 1 Introduction

In bush environments, where often under-manned personnel including bush firefighters and search-and-rescue teams work and could greatly benefit from robotic assistance, the task of navigation is often very difficult. Because such environments *are* the workplace of a number of groups, they invariably contain access tracks – whether vehicular dirt roads or smaller walking paths – that can and should be made use of whenever possible, as, by definition, they represent a relatively clear passageway between locations that have been determined to be of interest. While cross-country traversal may at times be necessary, the probability of mission success will be greater if tracks are utilised.

Additionally, many if not most of the larger paths and tracks in National and State Parks throughout Australia have already been mapped with these maps being available either commercially [Feathertop Mapping Services, 2006] or through government agencies [Parks Victoria, 2007]. Accordingly, it is a relatively easy task to find a high-level path between two points of interest that makes use of such tracks. This leaves the problem of how a robot may be guided safely and efficiently along a given track, once it has been chosen.

While a number of different sensor system could be used to perform autonomous or semi-autonomous navigation along such tracks, passive vision offers a number of benefits. Firstly, the equipment is cheap, perhaps the cheapest of the sensors most likely to be useful (with sonar and lasers being the two main competitors). Given the typically stretched resources the relevant organisations (for example the CFA and SES), low cost is important. Secondly, the natural world is visually rich, meaning there is a wealth of visual information that has the potential to be exploited. Finally, the ability to make use of human perception and insight into the problem makes vision an attractive choice in such an unstructured and variable environment.

The work presented in this paper aims to tackle a particular element of the wider navigation problem. Specifically we investigate the coupled problems of firstly assessing the traversability of the local terrain and secondly finding a short-term path, relative to the robot, that takes the robot along a track while avoiding obstacles. That is, the path determined must be efficient, by staying on the track when possible, and it must be safe, by avoiding any form of obstacle that it is likely to encounter. We choose to attempt this using passive vision alone, motivated by the reasons stated earlier. The solution presented uses stereo vision to find obstacles, colour image processing to characterise and extract the track surface, and simple path-planning techniques to combine these two classes of information and determine a safe path along the track.

## 2 Related Work

The work presented in this paper formed part of the author's doctoral dissertation [Fernandez, 2007] and extends and improves upon earlier work presented in [Fernandez and Price, 2005]. Accordingly, a thorough review of related works can be found in those publications. In summary, we note that the most closely related work is found in road following projects. Sealed urban roads have been the primary target of autonomous road-followers (as indicated by [Bertozzi *et al.*, 2000; Bertozzi *et al.*, 2002]), where assumptions of consistent geometry and visible road markings can and have been used, often with great success. In comparatively degraded bush environments, where structure is minimal at best, such assumptions simply do not hold. When poorly structured roads have been the focus, colour vision is the most common sensing modality [Thorpe *et al.*, 1991; Crisman and Thorpe, 1991; Fernández and Casals, 1997; Ghurchian *et al.*, 2002] though work has been done where colour vision is combined with other sensors, such as lasers [Rasmussen, 2002]. In most cases, the inherent difference in colour of road and non-road regions, typically the result of a difference in surface materials, is exploited, allowing road-surface regions to be characterised and hence extracted.

## 3 A System-level Overview

The task of autonomously navigating along bush tracks is broken down into three sub-problems: determining the presence and location of obstacles, determining the location of the track, and determining a local path that will safely guide the robot, staying on the track as much as possible while avoiding obstacles.

Here obstacles are defined as any region of space that could serve to impede or hinder the robot's motion. As such, we are not necessarily considering distinct or discrete objects; a particularly bumpy and uneven patch of ground is as much of an obstacle as a fallen tree and both should be avoided to minimise the chance of becoming stuck or being damaged. This classification of the environment, the "*obstacle map*", describes the areas we wish to avoid to ensure safety.

As described in earlier work [Fernandez and Price, 2005; Fernandez, 2007], the track is characterised and defined in terms of its colour-space statistics, following the assumption that the track surface is visually different to the surrounding environment, primarily because of a difference in surface materials (for example, brown dirt or grey gravel compared to green surrounding foliage). Using a dynamic description of the track that adapts to the current terrain, the environment is classified based on its similarity to the current track description. This second classification of the environment, the "*road map*", describes the areas we wish to traverse over in order to afford an efficient path.

The combination of these two maps gives rise to what we call the "*amenability map*", a description of the terrain's amenability to safe traversal, given the desire to stay on tracks where possible while avoiding obstacles. This map, along with a set of geometric constraints, is then used to determine a local path that begins at the robot and ends on the track some distance away, avoiding obstacles and staying on the track along its trajectory.

## 4 Obstacle Space

The obstacle detection system uses a calibrated stereo-camera-pair, positioned to give a view of the terrain in front of the robot that provides a depth-map of the terrain. We define an obstacle as any region of space that deviates from an assumed ground plane. This allows both positive obstacles – those that extend *above* the ground such as rocks and fallen branches – and negative obstacles – those that extend *below* the ground such as potholes and ruts – to be detected equally. The ground plane is itself dynamically defined, that is, it is determined from the current depth-map, allowing local variations in the terrain, such as slight inclines or declines, to be accommodated.

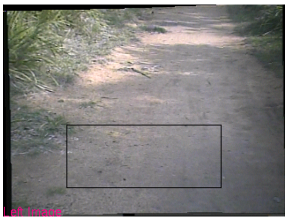


Figure 1: Sampled ground region

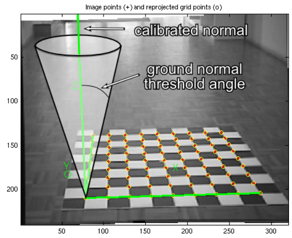


Figure 2: Ground normal constraint cone

The fitted ground plane is described by a (unit) normal vector $\hat{\mathbf{N}}_p$ and an anchor point $\boldsymbol{A}_p$. Any three (non-collinear) points $\boldsymbol{p}_a$, $\boldsymbol{p}_b$ and $\boldsymbol{p}_c$ can be used to define the plane as follows:

$$\hat{\mathbf{N}}_p = \frac{(\boldsymbol{p}_b - \boldsymbol{p}_a) \times (\boldsymbol{p}_c - \boldsymbol{p}_a)}{\|(\boldsymbol{p}_b - \boldsymbol{p}_a) \times (\boldsymbol{p}_c - \boldsymbol{p}_a)\|} \tag{1}$$

and

$$\boldsymbol{A}_p = \frac{\boldsymbol{p}_a + \boldsymbol{p}_b + \boldsymbol{p}_c}{3} \tag{2}$$

A window of depth-map points is sampled and the RANSAC algorithm [Fischler and Bolles, 1981] is used to find and fit a plane to that region, assuming that most of what is visible in the sampled region is clear ground. The windows is shown in Figure 1, superimposed on the raw stereo image for clarity.

To ensure only sensible planes are used a normal constraint is included which is simply an upper threshold on the allowed angle between a recovered normal and a calibrated normal, where the angle between two (unit) normal vectors is defined by

$$\theta = \mathrm{Cos}^{-1}(\hat{\mathbf{N}}_{cal} \cdot \hat{\mathbf{N}}_{est}) \tag{3}$$

Essentially, the calibrated normal and the threshold angle, $\theta_{norm}$ describe a cone (shown in Figure 2) and for a plane to be considered viable its normal must lie within this cone.

Creation of the obstacle map consists of a mapping of the absolute height of terrain points compared to

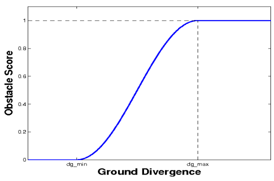


Figure 3: Obstacle mapping function

the ground. Using the minimum distance, $h$, between the point $\boldsymbol{p}$ and the plane:

$$h = (\boldsymbol{p} - \boldsymbol{A}_p) \cdot \hat{\mathbf{N}}_p \tag{4}$$

we then define the ground divergence $d_g$ of a point as

$$d_g = |h| \tag{5}$$

The ground-based obstacle likelihood $O_g$, ranging from 0 (clear ground) to 1 (highest likelihood of an obstacle) is defined as

$$O_g(d_g) = \begin{cases} 0 & , d_g < d_{floor} \\ \frac{1}{2} - \frac{1}{2}\cos\left(\frac{\pi(d_g - d_{floor})}{d_{ceil} - d_{floor}}\right) & , d_{floor} \leq d_g < d_{ceil} \\ 1 & , d_g \geq d_{ceil} \end{cases} \tag{6}$$

where $d_{floor}$ is a minimum divergence, below which points are assumed to be safe and hence given a zero obstacle score, and $d_{ceil}$ is a maximum divergence, above which divergences carry maximum penalty (greatest likelihood of an obstacle). Figure 3 depicts this relationship graphically.

The specific form of this mapping function is not critical. The key elements are an increase in likelihood of an obstacle with increasing divergence from planar and a maximum allowable divergence. The idea is that, above this threshold, it no longer matters how much of divergence there is, it is simply enough to assign the highest likelihood of there being an impediment to motion. The divergence thresholds used are $d_{floor} = 25$, $d_{ceil} = 100$, empirically determined.

As a final point it should be noted that the default map state is one of high obstacle likelihood, that is, a pessimistic approach that errs on the side of caution. Accordingly, if the stereo-processing yields no information (typically because no correspondence match of high enough quality is found), we assume maximum likelihood.

## 5 Road Space

The road is defined by a set of dynamically extracted colour-space descriptors. An image region that is assumed to contain the road is repeatedly sampled and colour-space statistics are retrieved. If the retrieved descriptor differs enough from the descriptors already in the set, indicating the surface has changed, the new descriptor is added to the set and the descriptor with the lowest "utility" – a measure of usefulness related to the age of the descriptor and its ability to consistently describe the road – is discarded. In this manner, the system can readily adapt to changes in the road surface but still maintains an efficiently small set of descriptors. Each descriptor in the set is then used as the kernel of a colour filter, allowing each pixel to be given a likelihood of being part of the road based upon its similarity to the stored descriptors.

With variations in lighting being one of the larger problems faced outdoors – especially mixtures of strong sunlight and dark shadows – a colour representation that separates brightness or intensity from the "colour" information was used. The specific transform from a standard red, green, blue triplet $\boldsymbol{p}_{RGB} = [r, g, b]^T$ to a so-called XYI triplet $\boldsymbol{p}_{XYI} = [x, y, i]^T$ is given as follows:

$$\begin{bmatrix} x \\ y \\ i \end{bmatrix} = \begin{bmatrix} \frac{1}{6}(3+\sqrt{3}) & -\frac{1}{6}(3-\sqrt{3}) & -\frac{1}{\sqrt{3}} \\ -\frac{1}{6}(3-\sqrt{3}) & \frac{1}{6}(3+\sqrt{3}) & -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \tag{7}$$

with conventional normalisation to give in inputs $r, g, b \in [0, 1]$ and transformed values $x, y, i \in [0, 1]$.

Using the pixels within the sampling region, a modified K-means algorithm [Lloyd, 1982] is used to extract at most $K$ clusters, each giving a mean $\bar{c} = [\bar{c}_x, \bar{c}_y, \bar{c}_i]$ and a standard deviation $S = [S_x, S_y, S_i]$. The road-sampling window contains $N_{total}$ samples (pixels) that are distributed among $\hat{K}$ clusters (the number of clusters actually found and which is equal to or less than the $K$ clusters requested). With a uniform distribution of samples, we would expect a quota of $\frac{N_{total}}{\hat{K}}$ samples to be assigned to each cluster. A useful measure of confidence is the ratio of samples actually assigned to a cluster against this "uniform-assignment" quota, defined as follows:

$$\zeta = \frac{N \cdot \hat{K}}{N_{total}} \tag{8}$$

It is a description of how much data was used to produce a descriptor compared to an even spread of samples. Each road descriptor is then defined by a cluster mean $\bar{c}$ and standard deviation $S$, the number of samples within the cluster $N$ and a confidence measure $\zeta$. When used as a filter kernel, the total number of hits $h_{total}$ (the total number of pixels associated with this descriptor) and the descriptor's age $t$ (the number of processing iterations since the descriptors creation) are also stored.

To determine the likelihood that a pixel in the image is part of the road, we then perform a filtering operation, comparing each pixel against all of the stored road descriptors. For a given pixel of colour $[x, y, i]$, we define its score $\rho$ against a filter using tolerances of $\Delta = [\Delta_x, \Delta_y, \Delta_i]$ as follows:

$$d = \sqrt{\left(\frac{x - \bar{c}_x}{\Delta_x S_x}\right)^2 + \left(\frac{y - \bar{c}_y}{\Delta_y S_y}\right)^2 + \left(\frac{i - \bar{c}_i}{\Delta_i S_i}\right)^2} \tag{9}$$

and

$$\rho = \begin{cases} minimum\big(\, 1,\ (1-d)\cdot\zeta \,\big) & ,d \leq 1 \\ 0 & ,otherwise \end{cases} \tag{10}$$

where $\zeta$ is the road descriptor confidence, used as a weighting factor. Note also that the score $\rho$ is limited to the range $[0.0, 1.0]$. Typical tolerances, empirically determined, are $\Delta_x = 3.0$, $\Delta_y = 3.0$ and $\Delta_i = 4.0$, reflecting the desired greater tolerance to variations in intensity compared with variations in colour. With the use of multiple descriptors and hence multiple filters, we simply assign to each pixel the highest score it receives, incrementing the hit-count $h_{total}$ of the associated filter in the process. We now have a way of determining the likelihood of a pixel belonging to the road, based on its similarity to a sampled road section.

In order to adapt to the current environment and changes in the appearance of the road (for example, changes that occur when moving through shadows or from dirt onto gravel), the system samples the road in order to create a new colour filter. However, for much of the time, we can expect there to be only small changes in the appearance, essentially caused by the specific patch of road that is being sampled. To avoid the inefficient creation of new filters that do not actually help (because the colours they allow are largely already covered by existing filters), we introduce a means of comparing a new descriptor against those of the existing filters.

For two descriptors and a tolerance $\Delta = [\Delta_x, \Delta_y, \Delta_i]$ we first define the following quantities:

$$\begin{aligned} x_{0,min} &= \bar{c}_0 - \Delta_x S_0 \\ x_{0,max} &= \bar{c}_0 + \Delta_x S_0 \\ x_{1,min} &= \bar{c}_1 - \Delta_x S_1 \\ x_{1,max} &= \bar{c}_1 + \Delta_x S_1 \\ X_{min} &= maximum(x_{0,min}\ ,\ x_{1,min}\ ) \\ X_{max} &= minimum(x_{0,max}\ ,\ x_{1,max}\ ) \end{aligned}$$

We then define the overlap of the two filters in the X channel as follows:

$$X_{0,overlap} = \frac{X_{max} - X_{min}}{2\Delta_x S_0}$$

with $Y_{0,overlap}$ and $I_{0,overlap}$ defined similarly for the remaining two colour channels. These measures describe the amount of overlap between two filters in each colour channel, normalised against the spread of the first filter. (It should be noted that this method of calculating overlap implicitly treats the filters as describing a cuboid, not ellipsoid, region of colour-space, and so is a simplified approximation to the true overlap.) A positive overlap indicates that there is a shared region in the relevant channel, while a zero or negative overlap indicates no common coverage. We then define the similarity of the two filters, referenced against the first, as follows:

$$s_0 = minimum(\ X_{0,overlap},\ Y_{0,overlap},\ I_{0,overlap}\ ) \tag{11}$$

Two filters that share a common region in colour space (under the cuboid approximation) will give a positive similarity, while two that do not (even if there is overlap in one or two channels) will give a zero or negative similarity. A new filter is only created if the similarity thus defined is less than an empirically determined threshold (a threshold of 0.9 is used), meaning the region in colour space covered by the new filter is not effectively covered by an existing filter.

Filters are held in a list that is sorted based on the filter's utility $u_F$, defined as follows:

$$u_F = \frac{h_{total}}{1+t} \tag{12}$$

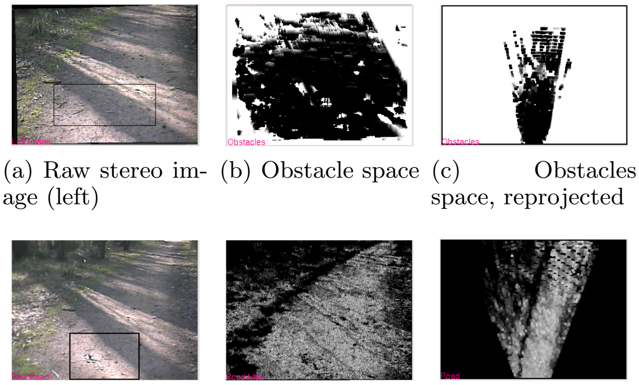
with $h_{total}$ and $t$ being the total hit count and age of the filter respectively. The utility is an indication of how useful, on average, a filter has been, with regards to its ability to consistently describe the road.

When a new filter is created (because the new statistics are different enough from all existing filters), the list of filters is first sorted by filter utility, the least useful filter is discarded and the new filter is added to the list. In this way, we provide adaptability to current conditions (through the creation of new filters) and a manageable temporal accumulation of knowledge (through a fixed-size set of filters that is modified as time passes).

## 6 Finding an Amenable Path

We now have available two environmental maps, one describing the likelihood of an area being part of an obstacle and another describing the likelihood of an area belonging to the road. While at first glance it might seem that these two maps are simply inversions of one-another, this is not necessarily the case. For example, a rut or a particularly bumpy and uneven patch of the track would both most likely still look like the track and hence be given a high likelihood of being part of the track. According to our earlier definition of what constitutes an obstacle, however, these regions are unfavourable and should be avoided, and would be given a moderate to high likelihood of being part of an obstacle. In another case, we may wish – or need – to deviate off the track, perhaps to move around a large obstacle (such as a fallen tree), in which case we still wish to stay where the ground is obstacle-free, but not necessarily stay on the track. As such, the two likelihood maps are independent but compatible. The combination of these maps describes the environment in terms of overall amenability to traversal.

In the physical configuration used, the camera used for finding the track is *not* either of the stereo cameras and as such the road map and obstacle map are in different frames of reference. Image registration through



(a) Raw stereo image (left) (b) Obstacle space (c) Obstacles space, reprojected

(d) Raw road image (e) Road space (f) Road space, reprojected

Figure 4: Re-projection of obstacle and road spaces onto virtual camera

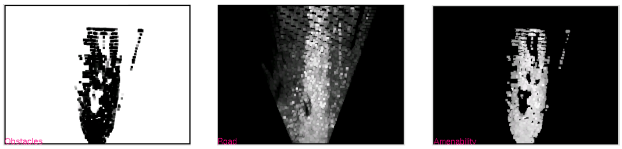
re-projection onto a virtual camera (that has a birds-eye view), using data obtained during initial system calibration, is therefore the first step in combining the two maps. The results of such a re-projection are shown in Figure 4. In these and similar figures, high likelihoods are shown as light regions and low likelihoods are shown as dark regions.

The likelihood of a location having feature $f$ (where $f$ is either obstacle or road), is denoted as $L_f$ and is in the normalised range $[0, 1]$. However, we wish to provide a means by which the relative importance of each feature can be adjusted. Typically, we wish to place a slightly higher importance on avoiding obstacles, as traversal into an obstacle is likely to present a greater impediment to motion (and present a greater risk of mission failure) than simply straying off the road temporarily. However, under certain circumstances (for example, if we are required to travel cross-country, that is, off the road), we would wish to ignore the road information. Each feature is therefore assigned a simple weight, denoted $\gamma_f$, and the overall amenability to traversal, $A_{trav}$, is given by

$$A_{trav} = \gamma_{obs} L_{obs} + \gamma_{road} L_{road} \tag{13}$$

Accordingly, $\gamma_{obs}$ will be negative and $\gamma_{road}$ will be positive. An amenability below zero then indicates terrain we should strongly avoid, zero indicates somewhat neutrality with regards to traversal, and positive indicates the terrain we most strongly wish to traverse. Applying Eq$^n$ 13 to each point in the re-projected likelihood maps produces an overall amenability map, which we can then use to deduce a safe path. Figure 5 shows an obstacle and a road likelihood map, and the resultant amenability map.

The amenability map is in essence a finely-grained occupancy grid where the value of each cell is the likelihood of that cell being safely traversable. This is considered to be both a feasible and effective representation because the map is only intended to be kept for a short time and used locally, and so problems associated with scaling (and the corresponding increases in storage and processing requirements) are not likely to occur.



(a) Obstacle-space (b) Road-space (c) Amenability

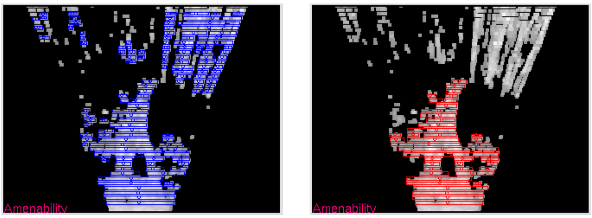
Figure 5: Obstacle and road likelihood maps and the resultant amenability map

Having determined for each point in the map an amenability to traversal, we are now in a position to find potential paths, compare them, and determine the best candidate, which then determines where the robot's next waypoint will lie. This process consists of three steps: finding contiguous regions in the map, linking them to form potential candidates, and fitting a model to each candidate to determine its suitability.

Traditionally, maps based on occupancy-grids are typically coupled with planners based on the distance-transform or potential-field concepts, where the path from known starting location (that is, the current position) to known destination is incrementally built, for example, by performing a steepest descent operation using the distance transform or by iteratively determining the resultant "force" being applied to the robot by the attraction of a goal and the repulsion of obstacles in a potential field. In either case, the destination must be known (whether provided or determined) before a path can be planned. In the context of following dirt roads or attempting straight-line cross-country traversal, these intermediate goals must be determined from an examination of the locally produced amenability map, applying the high-level goal of either following the road or maintaining a bearing, to find a destination that lies within safe (that is, amenable) space and is in the appropriate direction (either the direction of the road or the desired bearing), but is located some distance away. The tasks of determining an intermediate destination and finding a path to that destination are therefore tightly coupled. Accordingly, a different approach to finding a path is adopted where no specific destination is used. Instead some simple geometric and environmental assumptions are made which allow the generation of multiple paths which can then be compared to find what is considered the best option.

### 6.1 Sliced Segments

With our underlying goal being to drive along a road, the assumption is made that the road we wish to follow, and hence the path we will wish to take, flows generally away from the robot in approximately the direction it is facing. This assumption, which will typ-



(a) All segments

(b) Merged and culled segments

Figure 6: Segments: before and after merging and culling

ically be valid whenever the robot is on the road and either already travelling along it or at least facing the direction it wishes to travel, allows the process of determining the path and intermediate goal to be simplified.

Following this assumption, the map of amenable terrain is separated into slices: horizontal regions that lie perpendicular to the direction the robot is facing. We then segment the map into contiguous regions within each slice, looking for what should be safe regions of terrain that may be separated by potentially unsafe areas. Segments are described by their centre-of-mass, total mass and area, where member pixels contribute according to their amenability (that is, the total mass is the sum of all contributing pixels' amenabilities, and the centre-of-mass is the amenability-weighted mean position of contributing pixels). Only segments above a minimum size (in both area and mass) are considered further. Within a slice, segments that are separated by less than a maximum distance are merged to allow for small discontinuities that should not impede motion. For example, a small tuft of green grass in the middle of a brown dirt road would not impede motion, but would likely be considered to not belong to the road, given the difference in appearance. Additionally, starting closest to the robot, the adjacency of segments between slices is examined. A gap between segments is not allowed, and any segment that is not directly adjacent to another segment that is closer to the robot is culled.

Figure 6 shows an example of a segmented amenability map. In Figure 6(a), all segments are shown, each described by a bounding rectangle and a circle at its centre of mass. Figure 6(b) shows the merged and culled segments, described similarly. As can be seen, many segments in the upper part of the image that are clearly not part of the road have been removed due to their separation from the main road body. Along the main road body, a number of small segments can also seen to have been merged with horizontally adjacent segments when the separation is very small.

### 6.2 The Graph of Good Ground

To maintain a safe path, we wish, as much as possible, to stay on or in the regions we know to be safe and amenable to motion. Other than the assumption that the road flows generally away from the robot, we do not know where it lies. It may veer to the left, it may veer to the right, it might fork or it may be interrupted by obstacles. To accommodate this, we create a graph from the derived segments. The graph is rooted at the robot, and extends away, linking increasingly distant segments that touch but are within adjacent slices. In the case of a fork, for example, we will have a single segment in the slice at the base of the fork but two, horizontally separated segments in the next slice, where the road has forked. The segment at the base will then be linked separately to the segments in both arms of the fork, and so on. Similarly, an obstacle in the middle of the road will produce a graph that forks around the obstacle and recombines beyond it. The centre-of-mass of each included segment forms a node in the graph, such that the graph flows between points of maximum amenability (following the assumption that the centre-of-mass is the most amenable point within a segment). By construction, the graph is directed and acyclic, as each new node that is added may be the child of multiple existing nodes, but is the parent of none.

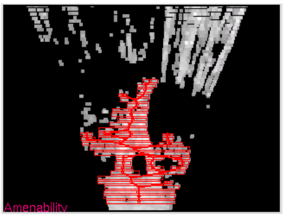


Figure 7: The path graph

This graph represents the set of potential short-term paths that conform to our requirement of staying on safe terrain. The adjacency requirement means a robot moving along the graph should never travel over unsafe terrain.

Figure 7 is an example path graph, showing path segments (as previously) as the resultant graph that connects the segment centres-of-mass.

### 6.3 From a Graph to a Trunk: Evaluating Potential Paths

The graph that we have created describes, in general, a whole series of potential paths, any of which may be taken. We must therefore reduce this graph to a single trunk that represents the desired path. To do this, we need a way of comparing potential paths against eachother, according to some metric, allowing them to be ranked and a most favourable candidate to be chosen.

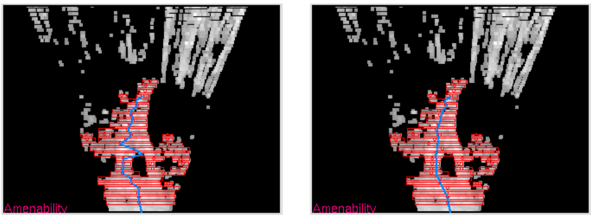
As much as possible, for both simplicity and efficiency, we would like the robot to take a smooth path that takes the robot as far along the road as possible. By smooth we mean a path that will require only small steering changes, reducing the chance of errors associated with large or rapid changes in the direction of the robot. The desire for a relatively long path trades off the ability to react to environmental changes against the efficiency of needing many (computationally) costly scanning and planning operations. In the bush, it is assumed that the physical environment is largely static. While wind can cause rapid motion of branches and shifting clouds can quickly change lighting and shadows, for example, these changes do not have a significant effect – if any effect at all – on where it is safe to travel. Moving obstacles are also unlikely to be encountered, largely because human and vehicular traffic is very low in these areas. Those obstacles that are likely to be encountered – animals, such as kangaroos – typically move faster than the robot could realistically react to, and so an attempt to accommodate them would lead to ineffective planning.

We would also like the robot to travel where it is considered safest, meaning traversing a path that is highly amenable (according to obstacle and road detection) and that provides physical safety by being wide and offering large buffers on either side of the robot.

To find a potential path, we begin at the root of the graph and work outwards, performing a breadth-first traversal. At each stage, the current node is appended to a list of nodes that describes the path taken to reach the current node. It is this list of nodes (where each node is a segment in the amenability map) that is used to fit a path model for evaluation. A limit is placed on the total number of paths allowed to be evaluated, in order to place an approximate bound on the processing time required when the amenability map is poorly constructed and produces a dense, highly-connected graph. By using a breadth-first traversal, we favour short-distance, wide-area exploration of paths instead of long-distance, narrow-area exploration, as would be obtained with a limited depth-first traversal.

To create a smooth path, a small-window smoothing operation is first performed on the centres-of-mass of the contributing segments. This is primarily done to remove large jumps between adjacent segment centres-of-mass that typically occur when a narrow segment appears adjacent to a wide one. For example, at the point of splitting, a fork in the road will produce two segments that will each be approximately half the width of the preceding segment. This is depicted in Figure 8, where segment centres-of-mass are shown as circles and the trace of contributing segments is shown as a solid line. In Figure 8(a) the trace undergoes a sharp change in direction at the point of forking, while in Figure 8(b), with smoothing, the change in direction is much less extreme. In this usage, smoothing serves to avoid potential problems associated with rapid changes in robot direction and the potentially dangerous situation of cutting corners.

We choose to model a potential path as a simple second-order polynomial, reflecting the desire for a path that can accommodate road curvature yet still produces a smooth trajectory. In fitting the polynomial to a potential path we use the error – the difference between the model and the contributing nodes' locations – as an indication of how smooth that path is (compared to the idealised polynomial path).



(a) Un-smoothed (b) Smoothed

Figure 8: Original and smoothed potential path

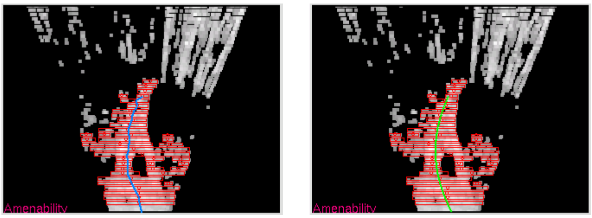
There is one other factor that needs to be considered when selecting a final path: direction. In some situations, most notably when facing a fork in the road, a decision needs to be made (typically by a higher-level, global path-planner) as to which fork to take. At other times, the robot may be traversing cross-country, and needs to maintain a given bearing, again provided by a higher-level planner. In the more general case, we will simply wish to maintain a consistent trajectory and provide some immunity to instabilities and vacillation that may occur when the choice of path is not distinct. To allow for these types of control, we introduce the notion of path bearing. In evaluating a path, in addition to the second-order polynomial, a first-order polynomial is fitted, and the gradient of this model is used to determine the overall bearing of the path, relative to the robot. We then have a form of control with the bearing error, equal to the absolute difference between the path bearing and the desired bearing. The desired bearing is either given by a global-planner or simply equal to the previous bearing in order to produce a temporally consistent choice of path.

For a given potential path, we have a number of defining characteristics. The total amenability $A$ is the sum of all contributing segment masses. The displacement $d$ is the straight-line distance between the first and last contributing segments. The error $\epsilon$ is the difference between (smoothed) segment centres-of-mass and the fitted model. The minimum buffer $b$ is the minimum distance between the fitted model and horizontal segment boundaries. The average width $\bar{w}$ is the average width of contributing segments. The bearing error $\theta$ is the absolute difference between the approximate path bearing and the desired bearing.

The overall fitness of this path is defined by:

$$f = c_A A + c_d d + c_\epsilon \epsilon + c_b b + c_{\bar{w}} \bar{w} + c_\theta \theta \tag{14}$$

where $\{c_A, c_d, c_\epsilon, c_b, c_{\bar{w}}, c_\theta\}$ are weighting coefficients that describe the relative importance of each trait. Typically we use $c_A = 1.5$, $c_d = 2.0$, $c_\epsilon = -0.5$, $c_b = 1.5$, $c_{\bar{w}} = 1.0$ and $c_\theta = -1.0$. With these values, we favour a highly amenable, long path with large buffering, and weakly penalise paths with a large fit-



(a) Trunk path (b) Fitted model

Figure 9: The path trunk

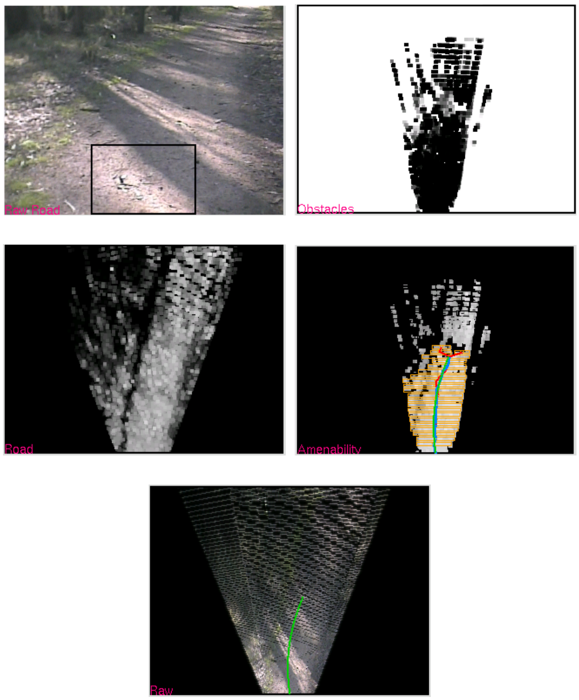


Figure 10: Road-following result 1

ting error, indicating a path that is not smooth, and paths that deviate from the desired bearing. Standard behaviour also simply sets the desired bearing to the previous bearing in order to provide a path that is temporally smooth.

## 7 Experimental Results

In this section a selection of local path-planning results will be presented in order to give an overview of the system's performance.

In the results that follow, the raw centre image, reprojected obstacle likelihood map, reprojected road likelihood map, amenability map (overlaid with segments, graph, trunk and fitted model) and reprojected centre image (overlaid with fitted model) are shown.

Figure 10 shows a fairly simple scene where the road veers off to the right, with the only potentially troublesome element being the alternating strong shadows and bright regions. The road map shows, however, that these potentially problematic lighting conditions are handled comfortably. There is a patch to the left of the road that is incorrectly considered to be part of the road, however, because of the brightly-lit leaf-litter's resemblance to the road surface. The ground is quite flat and the obstacle map reflects this, though there are some falsely classified obstacle regions in the distance. The amenability map accordingly shows these non-existent obstacles as regions to avoid. However, the overall shape of the safe region on the amenability map is a good representation of the environment, and the resultant path is a smooth curve to the right that follows the centre of the road.

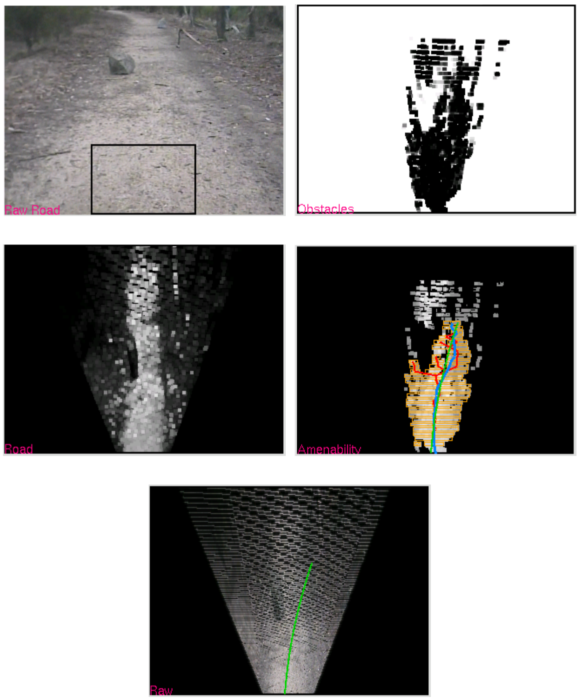


Figure 11: Road-following result 2

In Figure 11 a large rock is situated on the left side of the road surface, creating a clear obstacle. The obstacle map shows a safe and traversable environment except for the presence of the rock. In spite of there being a fairly small change in appearance between the sandy dirt road and the dry surrounds, the road map shows that the road surface has been very well extracted. The amenability map shows the terrain ahead to be quite safe ahead except for the large rock. The path chosen aims to veer around the rock and to the right while largely staying on the road.

In Figure 12 there is a very subtle obstacle infringing upon an otherwise clear road. There is a feathery overhanging branch that droops from above over the right side of the road. The colour of the branch is very similar to that of parts of the road making it essentially invisible in the road likelihood map (unlike some early obstacles, that had distinctly different appearances to the road). However, it is well detected by the stereo system and appears as a clear obstacle on the obstacle

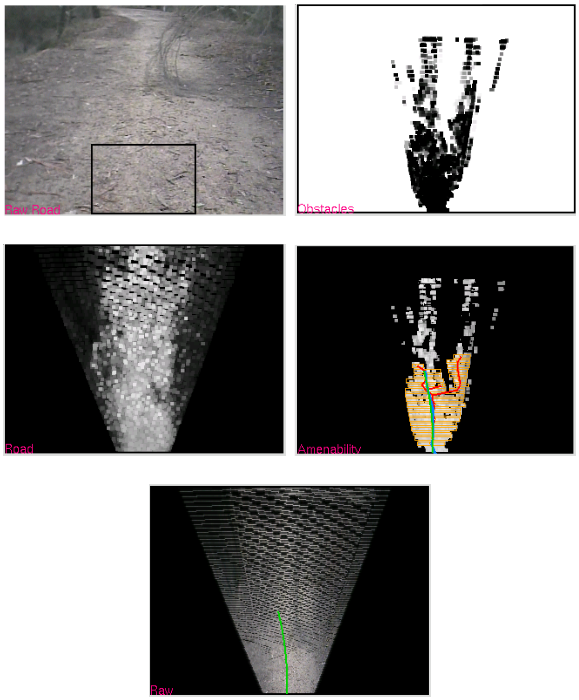


Figure 12: Road-following result 3

map. The chosen path, while stopping slightly short due to false-positive obstacles in the middle-ground, safely aims to guide the robot around and to the left of the branch, the safer of two alternatives (there is a potential path around and to the right, but the unobstructed road is wider to the left).

Some fallen branches lie on the left of the road in Figure 13. They are not at all distinctly visible on the road map, which does otherwise extract the road very clearly. The obstacle map, though, *does* show the presence of obstacles to the left on otherwise flat and clear ground. The generated path veers around and to the right of the branches, stopping short because of sparse stereo information in the distance and small regions of false-positives.

The next set of results show scenes of cross-country traversal. Accordingly, no road likelihood map is shown, and the amenability map is produced using only the obstacle likelihood map.

In Figure 7 the robot is facing short, step-like rise in the terrain with the ground quite flat on both sides step but at different levels. The obstacle map clearly shows the higher ground to be unsafe while the foreground and region to the right is flat and so considered safe. The path produced veers to the right, aiming to steer the robot away from the step and guide it parallel to it.

The scene in Figure 7 is quite cluttered on the left with wispy branches and a large tuft of grass. There is a fairly clear path around the shrubbery to the right, and this is firstly shown in the obstacle and amenability maps and secondly taken advantage of by the path-

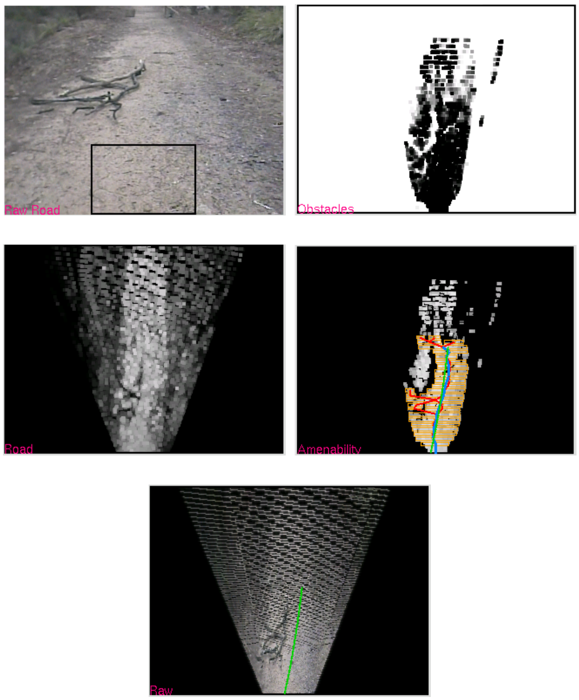


Figure 13: Road-following result 4

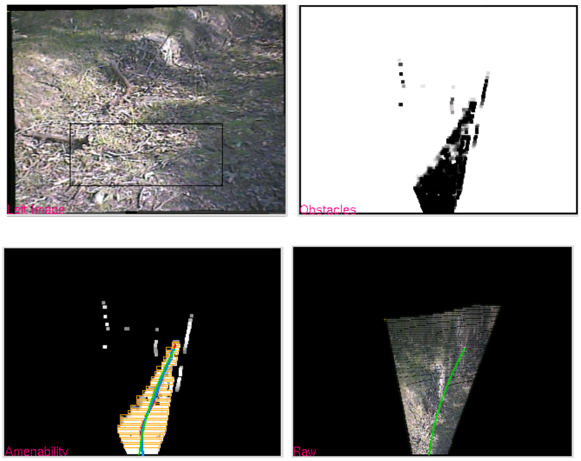


Figure 14: Cross-country result 1

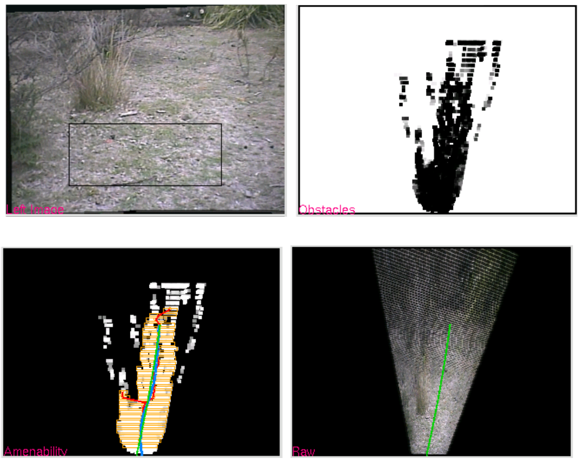


Figure 15: Cross-country result 2

planner.

## 8 Discussion

The results section above has shown the system's ability to evaluate the terrain under varied conditions and consistently determine which regions are likely to belong to the track and to contain obstacles and therefore which regions are likely to be amenable to safe traversal. Following this, the determination of an efficient local path that serves to take the robot along the track while avoiding obstacles is shown. Obstacles both obvious – such as large rocks and embankments – and more subtle – including overhanging and fallen branches – are detected, while extraction of the track is shown to be resilient to adverse conditions including strong sunlight and shadows and situations where the track is only slightly different in appearance to the surrounds. What has not been shown here is the dynamic behaviour of the system, in particular the possibility of the system, under certain circumstances, to enter a detrimental positive feedback loop. If the robot wanders off the track for too long, perhaps in order to avoid a large obstacle, it is possible for the history of track descriptors (Section 5) to be filled completely with "off-road" descriptors, causing the system to then try to stay *off* the track. This is largely a result of the current system obtaining the obstacle and road maps independently. Future work aims solve this problem by using the obstacle map to guide the extraction of road descriptors: only regions free from obstacles would be used to characterise the road and so the likelihood of obtaining and retaining off-road descriptors is significantly reduced.

## 9 Conclusions

In this paper we presented a navigation subsystem that aims to autonomously guide a robot along a rough bush track. The aim is to determine where the track is, where any obstacles may lie, and then choose an appropriate path that stays on the track where possible while still avoiding obstacles. The novelty of the approach presented, in part, lies with the fact that this is done using *only* passive vision, making it a cheap and simple (in terms of equipment) solution that could be used on robots aimed to assist SES or CFA workers, for instance, who often find themselves undermanned and in life-threatening situations where helper-robots could be of great use.

## References

[Bertozzi *et al.*, 2000] M. Bertozzi, A. Broggi, and A. Fascioli. Vision-based intelligent vehicles: State of the art and perspectives. *Robotics and Autonomous Systems*, 32(1):1–16, Jul 2000.

[Bertozzi *et al.*, 2002] M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta. Artificial vision in road vehicles. *Proceedings of the IEEE*, 90:1258–1271, Jul 2002.

[Crisman and Thorpe, 1991] J. Crisman and C. Thorpe. Unscarf-a color vision system for the detection of unstructured roads. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2496–2501, Apr 1991.

[Feathertop Mapping Services, 2006] Feathertop Mapping Services. Feathertop and rooftop maps. http://www.feathertopmapping.biz/, 2006. Visited 2006.

[Fernández and Casals, 1997] J. Fernández and A. Casals. Autonomous navigation in ill-structured outdoor environment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 395–400, 1997.

[Fernandez and Price, 2005] D. Fernandez and A. Price. Visual detection and tracking of poorly structured dirt roads. In *12th International Conference on Advanced Robotics*, pages 553–560, 2005.

[Fernandez, 2007] D. Fernandez. *Visual Navigation on Rough Tracks in Bush Environments.* PhD thesis, Intelligent Robotics Research Centre, Department of Electrical and Computer Systems Engineering, Monash University, Australia, 2007.

[Fischler and Bolles, 1981] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[Ghurchian *et al.*, 2002] R. Ghurchian, T. Takahashi, Z. Wang, and E. Nakano. On robot self-navigation in outdoor environments by color image processing. In *7th International Conference on Control, Automation, Robotics and Vision*, volume 2, pages 625–630, Dec 2002.

[Lloyd, 1982] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28:129–137, 1982.

[Parks Victoria, 2007] Parks Victoria. Parkweb, 2007. Visited 2007.

[Rasmussen, 2002] C. Rasmussen. Combining laser range, color, and texture cues for autonomous road following. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 4320–4325, 2002.

[Thorpe *et al.*, 1991] C. Thorpe, M. Herbert, T. Kanade, and S. Shafer. Toward autonomous driving: the cmu navlab. i. perception. *IEEE Expert*, 6(4):31–42, 1991.