Visual Detection and Tracking of Poorly Structured Dirt Roads

David Fernandez and Andrew Price
Department of Electrical & Computer Systems Engineering
Monash University
Clayton, Victoria 3168, Australia
David.Fernandez@eng.monash.edu.au

Fig. 1. A typical dirt road

Abstract— **Outdoor mobile robots are often faced with the problem of trying to navigate through an unknown environment. Areas that appear simple to humans can be very difficult for a robot to accurately and consistently describe. Poorly structured dirt roads, such as fire-access tracks and bush-walking tracks, are often overlooked in research but are highly important passage-ways for emergency support crews, such as search-and-rescue teams and firefighters tackling bushfires. This paper presents a method of autonomously detecting and hence tracking such roads using colour vision. Central to the process is a method of characterising the road surface through a statistical colour description, which makes minimal assumptions about the road. A highly simplified and generalised road model is used to ignore the background and contain the road, and weighted control points are used to generate a spline-based trajectory along the road, which is intended to be used for motion-control of a robot trying to traverse these tracks. Inherent to the system is the avoidance or safe traversal of certain types of obstacles. The combination of simple modelling and efficient processing algorithms has resulted in a usable average processing speed of approximately eight frames per second on the 1.7GHz Pentium-4 test machine.**

I. Introduction

The intention to have robots assist in human operations is a central notion in robotic research and development. This can mean many things, from ensuring their interfaces are simple and intuitive to providing a means for them to manipulate objects made for fingers and hands. It also means that, to be successful and practical, a robot needs to operate in the same environments as humans. An important aspect of this is the ability to competently navigate through areas used by people, especially areas used by humans for transport. Predominantly this means varying forms of roads, including but not limited to, paved footpaths, sealed multi-lane roadways and less-often-used but no less important unsealed and poorly structured roads and paths. Roads such as these may not see the heavy traffic of city-linking arterials, but in certain circumstances their availability can mean the difference between life and death: the presence of fire-access tracks through bush-lands, for example, can allow deadly bushfires to be controlled and contained, potentially saving many lives.

The research contained herein forms part of a larger outdoor robotics project that aims to develop a system capable of augmenting or performing support duties for human emergency response teams. An example of such duties would be the gathering and relaying of environmental conditions to a team of otherwise preoccupied fire-fighters. Another use would be in assisting search-and-rescue personnel when looking for lost or injured hikers and bush-walkers. If the robot is to be able to perform autonomously an ability to navigate and manoeuvre is paramount to successful operation. Previous work includes a visual odometry system [1] capable of tracking a robot's motion over a wide range of outdoor surfaces, including asphalt, grass, dirt, gravel and combinations of these. With the ideas presented here it will form the basis of a visual navigation system for an autonomous or semi-autonomous robot.

In the text that follows, the terms *road* and *track* are used interchangeably to refer to a poorly structured, unsealed, unmarked track whose surface consists primarily of dirt, rocks/stones, gravel, leaf-litter or other similar natural materials. These features are typical of fire-access roads and hiking tracks in hilly and mountainous forest regions around the world, such as those in south-eastern Australia. Figure 1 depicts a typical dirt road that is being considered.

II. Related Work

Sealed and marked roads, being much more common than what can be referred to as simply "dirt roads", have been thoroughly investigated, with many advancements having been made toward systems capable of autonomous navigation. Use has been made of geometric constraints [2], edge following [3], [4], [5], colour and pattern-recognition combined with geometry [6], and optical flow [7]. However, these methods typically take advantage of features or characteristics that are simply not present on dirt roads, with lane markings, distinct road edges and assumed geometry featuring extensively.

Research presented in [8] moves away from sealed, largely structured roads. It uses colour to distinguish the road from

0-7803-9177-2/05/$20.00©2005 IEEE

non-road areas with a change to an intensity-based method if the "colourfulness" of the area of interest (calculated through a saturation variant) is below a certain threshold. The assumed road area is only sampled once, meaning changes in the road appearance (such as common transitions through gravel, sand and mud or clay) cannot be accounted for. The deficiency of a method based on static description is shown where the test robot failed when "the road surface was completely degraded and covered by little stones". When using the colour-based method, evaluation utilised one of red, green and blue colour components only, meaning potentially useful information is being ignored. When using the intensity-based method, a parameter defined by $max(R, G) - B$ is used, based on (and hence limited by) the assumption that the road is largely blue (asphalt-based) and the surrounds are either green- (e.g. grassy) or red-based (dirt or clay, for example). Additionally, a road model is used that assumes straight road edges, which can make useful fitting difficult on poorly structured and maintained dirt roads, whose edges are rarely smooth let alone straight.

The work described here in part generalises and hence improves that in [8] through its relaxing of constraints and assumptions. There are only three key assumptions being made: that a small part of the image always contains the road (i.e., that the road to be detected and followed can be seen consistently), that the road surface appears *different* (in terms of colour description) to the surrounds (allowing it to be characterised and distinguished from the surrounds), and that the road to be followed flows away from the camera and hence appears in images to flow from the bottom to the top (so a simple road model can be fitted and a path created). Through its adaptive use of all colour information available it is robust against all but the most extreme lighting conditions and road appearances. Finally, it does not attempt to produce a detailed geometric model of the road; rather it provides a point-based trajectory along it which makes no further assumptions about the road geometry. This is intended to be used for reactive control to allow the detected road to be followed (for example, by controlling steering angles).

III. System Overview

The problem of detecting and tracking is first broadly divided into three tasks: definition of the road, clustering of road regions and modelling the trajectory of the road. The first step serves to provide an elementary partitioning of an image, ultimately dividing the image into low-level (that is, non-descriptive) areas that are considered to be either a part of or separate from the road. Having performed this operation, the areas classified as belonging to the road can be analysed and grouped, with further culling to remove "outliers", those being areas that appear similar to the road but are unlikely to belong to the major road area. Finally, properties of the remaining clustered regions are used to provide a basis for determining the trajectory of the road and hence a safe path along it. These three steps are essentially independent, so if an alternative method or algorithm is developed (for example, if a better road model were to be developed), it can replace the existing component with minimal modifications required.

Figure 2 gives a graphical overview of the process. The original image is shown in 2(a), followed by the result of colour filtering in 2(b). Figure 2(c) displays the raw slice segments, some of which (at the very top of the image) have been removed in 2(d), indicating areas appearing similar to the road but not belonging to it. Finally in 2(e) we see the original image overlaid with the remaining slice segments and the spline trajectory. Details of these processes are given in the following sections.

IV. Definition of The Dirt

In order to define the dirt road in a manner that allows it to be extracted and separated from its surrounds, certain assumptions need to be made. The prime assumption being made here is that the road surface displays colour-space statistics that are different to the surrounding regions. This allows an image of the road to be partitioned by filtering into pixels that potentially belong to the road (they are a similar colour) and those that do not (they are different enough to be ignored).

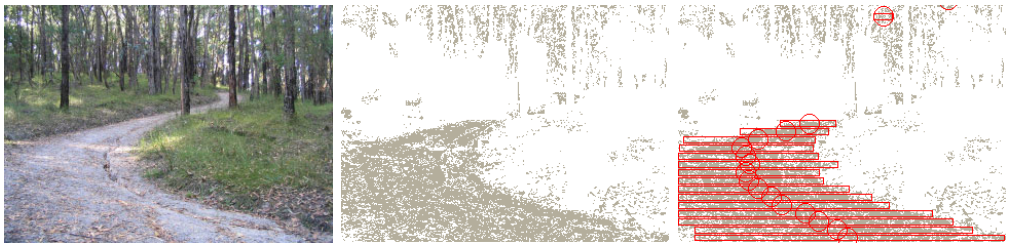
A. Colour Representation

A Hue, Saturation and Intensity (HSI) variant colour-space is being utilised for this first stage of defining the road and filtering out unwanted areas.

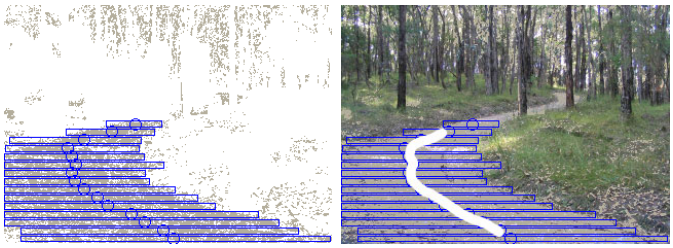
Tests were performed to compare three candidate colour models: RGB, HSI and YUV. A 48-image database was used for testing, and each image had a road and a non-road region manually marked (that is, a rectangle was marked that contained only the road and a second rectangle was marked that contained only non-road). The regions chosen were as large as practical while maintaining distinct road or non-road containment. These two regions were then statistically analysed in each colour-space. Three aspects were examined: histogram similarity (as measured through a normalised sum-of-absolute-differences) between road and non-road regions, road standard deviation and non-road standard deviation. These calculations were performed separately in each channel of each colour-space. The most useful colour space would give low standard deviations (indicating that road and non-road statistics are tightly clustered) and a low similarity (indicating discernibility between road and non-road statistics). The results of testing are summarised in Table I. In the results shown, similarity is normalised to the range $[0, 1]$ and all properties were analysed to the normalised range $[0, 255]$ (i.e. 8-bits).

As can be seen, HSI is consistently better than either RGB or YUV representations, making it a good choice for the task at hand. Of particular note is the significantly low similarity between road and non-road regions in the hue channel, supporting the notion that it can be used to distinguish road from non-road regions.

The specific transform from native Red-Green-Blue (RGB) colour-space to HSI being used differs from most commonly



(a) Original (b) Colour Filtered (c) Slice Segments



(d) Culled Slice Segments (e) Overlaid Trajectory

Fig. 2. Process Overview

Property	Similarity	Road σ	Non-road σ
Red	0.403	028.212	047.738
Green	0.583	027.490	050.141
Blue	0.451	026.579	049.589
Hue	0.155	034.479	047.824
Saturation	0.541	007.375	017.855
Intensity	0.471	026.933	047.637
Y	0.480	028.010	048.679
U	0.591	109.607	106.969
V	0.243	060.663	103.483
Average RGB	0.479	027.427	049.156
Average HSI	0.389	022.929	037.772
Average YUV	0.438	066.093	086.377

TABLE I
COLOUR-SPACE COMPARISON

encountered. Instead of the usual transform that uses maximums and minimums of channels and a hue calculation that is dependent on which channel is the maximum, the conversion used here follows from a simple coordinate transform that converts a Euclidean RGB triplet into a cylindrical-coordinate HSI point. The Intensity axis lies on the line $R = G = B$, Saturation is the radial distance perpendicular to the I-axis, and Hue is the angle measured around the I-axis. The transform is considered to be more mathematically rigorous and smooth and displays only a single discontinuity along the circular hue axis.

This transformation is easily achieved by first converting from RGB to an intermediate Euclidean UVW who's W-axis is aligned with the intended I-axis, followed by a simple Euclidean to Cylindrical coordinate conversion.

A generalised coordinate transform consisting of a rotation about a unit vector $\vec{\alpha}$ of angle θ can be represented as follows:

$$\vec{q} = R(\vec{\alpha}, \theta)\vec{p} \tag{1}$$

where

$$\vec{q} = [x_q, y_q, z_q]^T$$

contains the coordinates of the point in the new space,

$$\vec{p} = [x_p, y_p, z_p]^T$$

contains the coordinates of the point in the original space, and $R(\vec{\alpha}, \theta)$ is the transform matrix (detailed in equation 2).

To achieve the desired transformation, converting $\vec{p} = [r, g, b]$ to $\vec{q} = [u, v, w]$, we use

$$\vec{\alpha} = \left[\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0\right]^T$$

and $\theta \simeq 54.7°$, or more exactly, $cos\theta = \frac{1}{\sqrt{3}}$ and $sin\theta = \frac{\sqrt{2}}{\sqrt{3}}$. This results in

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \frac{1}{6}(3+\sqrt{3}) & -\frac{1}{6}(3-\sqrt{3}) & -\frac{1}{\sqrt{3}} \\ -\frac{1}{6}(3-\sqrt{3}) & \frac{1}{6}(3+\sqrt{3}) & -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \tag{3}$$

Finally,

$$\begin{aligned} h &= Tan^{-1}\left(\frac{u}{v}\right) \\ s &= \sqrt{u^2 + v^2} \\ i &= w \end{aligned}$$

$$R(\vec{\alpha},\theta) = \begin{bmatrix} \alpha_x^2 + (1-\alpha_x^2)cos\theta & \alpha_x\alpha_y(1-cos\theta) + \alpha_z sin\theta & \alpha_z\alpha_x(1-cos\theta) - \alpha_y sin\theta \\ \alpha_x\alpha_y(1-cos\theta) - \alpha_z sin\theta & \alpha_y^2 + (1-\alpha_y^2)cos\theta & \alpha_z\alpha_y(1-cos\theta) + \alpha_x sin\theta \\ \alpha_z\alpha_x(1-cos\theta) + \alpha_y sin\theta & \alpha_z\alpha_y(1-cos\theta) - \alpha_x sin\theta & \alpha_z^2 + (1-\alpha_z^2)cos\theta \end{bmatrix} \quad (2)$$

It should also be noted that conventional normalisation is performed to give inputs $R, G, B \in [0, 1]$ and transformed values $S, I \in [0, 1]$ and $H \in [0, 360)$.

B. Statistical Analysis

One of the key aims of this work is to develop a system that is capable of detecting a dirt road *without* human intervention. Therefore, the system needs the ability to characterise the road automatically. This is done through a fairly simple statistical analysis. The key assumption here is that the robot is always able to see part of the road it is intending to detect and follow. Specifically, it is assumed that a small rectangle at the centre-bottom of the image always contains part of the road (although the position of this area can be arbitrarily changed if more appropriate). It is this area that is analysed and taken to be characteristic of the road in general. By performing the analysis at regular intervals changes in the road (such as those due to a surface or lighting change) can be handled comfortably by the adaption of the road parameters to current conditions. It should be noted that, in the current implementation, analysis occurs every frame and all results are gathered under this condition. Processing speed could be greatly increased if analysis were performed less regularly, such as every second or third frame.

Analysis consists of calculating and collecting the mean and standard deviation for the hue, saturation and intensity of the pixels within the above-mentioned characterising rectangle. These parameters are then used to create a colour-based filter. Saturation and intensity can be treated like any other variable for which statistical data is required, but hue presents a problem. This is due to the circular nature of hue description. It is not enough to simply take the sum of all hues and divide by the number of samples. While this may technically give the mean, what is actually desired is a description of "where most of the pixels lie", in terms of hue. On the $0 - 360$ hue scale, a distribution where most samples are either small numbers close to 0 or large numbers close to 360 should give a mean somewhere near these two (ultimately equivalent) values and a fairly small standard deviation. Standard equations, however, would give a mean somewhere near 180 and a very high standard deviation, clearly a very poor indication of where most samples are located and how densely clustered they are. To overcome this problem, a "shifted cutpoint, multiple hypothesis" method is used. The standard cutpoint of hue is at 0 and 360, that is, a hue of or above 360 is "wrapped around" and becomes a value slightly above 0. By repeating standard calculations using multiple, shifted cutpoints, more representative statistics can be obtained. To do this, the mean and standard deviation are calculated for a given cutpoint (initially the standard $360|0$). The samples are then circularly shifted to create a new cutpoint and new statistics are calculated. For example, a shift of -90 moves the cutpoint to $269|270$, and values previously clustered around $360|0$ now appear clustered around $89|90$. This is repeated for a small number of evenly spaced shifts (for example, 3 iterations with shifts of 120). The statistics with the lowest standard deviation are taken to be most representative of the data and corrected (effectively reverse shifting the mean). While this results in parameters that are technically not the true statistics, they are a much better indicator of where and how widely the samples are clustered, which is the information sought.

C. Image Filtering

Input images are passed through a colour filter that is an ellipsoid in the HSI colour-space. The filter is parameterised in terms of a target colour (h_T, s_T, i_T) (the means as calculated above) and an allowable tolerance $(\Delta h, \Delta s, \Delta i)$ (multiples (typically between 1.0 and 2.0) of the standard deviations). A given input (h, s, i) that satisfies the filter equation (given in equation 4) is passed through the filter and is considered to be a candidate for inclusion in the road definition. It should be noted that the distance $h - h_t$ is the minimum of the two possible circular distances.

$$\left(\frac{h-h_T}{\Delta h}\right)^2 + \left(\frac{s-s_T}{\Delta s}\right)^2 + \left(\frac{i-i_T}{\Delta i}\right)^2 \leq 1 \quad (4)$$

A method of recursive subdivision is used to perform the actual pixel-by-pixel processing. The image is first divided into a small number of sub-regions. Each sub-region is processed coarsely, for example, one out of every eight pixels is converted from RGB to HSI and tested against the filter. Upon finding a match, the current sub-region is further subdivided, and each new, smaller sub-region is processed at a finer scale of granularity, for example, every fourth pixel. This continues until the subdivision would result in a sub-region that is smaller than a given minimum size (determined empirically). At this point, no further subdivision is allowed and every pixel within the sub-region is processed. The process is depicted in Figure 3, where 1 is the initial area, subdivided into four regions. Upon finding a match within sub-region 2 (within the shaded area), 2 is sub-divided. Similarly, upon finding a match in sub-region 3, 3 is subdivided. A match found in sub-region 4 results in a complete processing of that area, as it is below the minimum sub-region size. If no matches are made within a region, no further subdivision or processing is performed on that region. The aim is to trade off some of the speed increase of a simple blanket sub-sampling for increased detail. The usefulness of this method is based upon the premise that the pixels of interest, that is, those that will be passed by the filter, are typically found clustered together and separated by regions of no interest. As a result, large areas

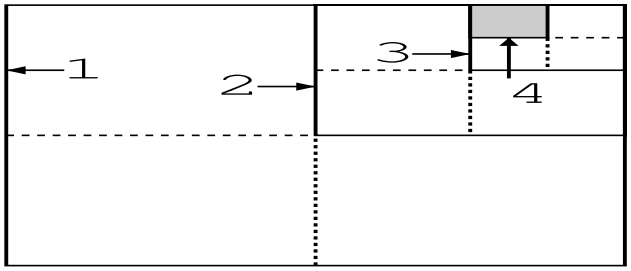


Fig. 3. Recursive subdivision for processing

Quantity tested	Result
Proportion of pixels tested [*rec-sub*]/[*std*]	0.671
Proportion of pixels passed [*rec-sub*]/[*std*]	0.980
Proportion of tested pixels passed [*std*]	0.404
Proportion of tested pixels passed [*rec-sub*]	0.588
Processing time (per image) [*std*]	7.2 ms
Processing time (per image) [*rec-sub*]	6.0 ms

[*std*]: standard method, [*rec-sub*]: recursive-subdivision method

TABLE II
PROCESSING METHOD COMPARISON

of the image that contain no (or very few) interesting pixels can effectively be skimmed over and processed very quickly. For the majority of images to be encountered, it is believed that the speed-up of reduced processing will far outweigh the overheads associated with subdivision and recursion. It should be noted, however, that by its very nature this method has a processing time that is highly dependent upon the image being processed. Additionally, because of the quantised (granular) processing, it is possible that some pixels of interest may be overlooked, particularly those that are isolated or in a very small group. However, this is not seen as detrimental, given the underlying desire to find *regions* of interesting pixels.

Tests have been performed to compare the recursive-subdivision algorithm to the standard all-pixels approach. Using a database of 48 road images, each image was filtered (using the colour-based filtering described) to produce a binary image. Each of these images was then processed using both methods, applying a simple threshold (essentially a true/false test) on each pixel being processed. This was done to minimise the effects of the individual pixel-processing operation. Testing aimed to determine a number of quantities: the relative number of pixels being tested by both methods (a measure of improvement), the relative number of pixels passing the filter (a measure of accuracy), the proportion of tested pixels passing the filter (a measure of efficiency) and the times taken to perform the operations (measures of improvement). Results (averaged over the image database) are shown in Table II.

The test results show a number of important things. Firstly, the proportion of tested pixels passed by the standard method, $\sim 40\%$, indicates that the image class being investigated typically contain pixels that are mostly *not* of interest. Possibly most importantly is the proportion of pixels passed. The recursive-subdivision method passes 98% of the pixels passed by the standard method, indicating the reduction in accuracy is negligible. The proportion of pixels tested by the recursive-subdivision method is $\sim 67\%$, meaning roughly a third of the image is being ignored by the recursive-subdivision method. This leads to the result of $\sim 59\%$ of tested pixels being passed, a 20% improvement in efficiency over the standard method. Finally, processing speed is reduced by about one millisecond (a $\sim 16\%$ reduction). Of note is the simplicity of the actual pixel-test (a threshold test). A more complicated test, where more time is spent testing an individual pixel, would clearly see an even bigger reduction in processing time, as the recursive-subdivision method processes far fewer pixels. Essentially, the more complicated the pixel filtering operation, the more beneficial the recursive-subdivision method becomes.

V. CULLING AND CLUSTERING REGIONS OF ROAD

At this point an image that contains only pixels that (may) belong to the road is available. The next step is to abstract individual pixels into regions that represent segments of the road. An assumption is made here about the road geometry. It is assumed that the road generally flows away from the camera and hence appears in the image to flow from bottom to top. Based on this assumption, the image is divided into horizontal slices that will determine slices of the road. A trade-off occurs here where tall slices give smooth trajectories because of fewer control points, but shorter slices give greater detail but potentially jagged trajectories as a result. Empirical testing found that 40 slices gave good results, although any number between approximately 20 and 60 worked well. Below this individual slices cover too much of the image and detail is insufficient. Above this, with 320×240 pixel images being used, the slices were too narrow to capture coherent regions. The number of slices used has a negligible impact on processing speeds (the difference in processing rate between 20 and 60 slices is less than one frame per second, with an average of eight frames per second being achieved), as approximately the same total number of pixels are processed. A series of region-growing segmentation operations are performed using the slices as boundaries, with multiple segments within each slice being allowed. Within each slice, segments whose bounding boxes are within a certain proximity of one-another are then merged. This is to allow for small discontinuities, such as small patches of grass or puddles that separate segments within slices. Once merging is complete, segments are culled, with only the largest segment within each slice being retained. Additionally, starting at the bottom-most slice, adjacent regions are compared for smoothness. A discontinuity between slices, that is, either a horizontal distance between adjacent bounding boxes being too large or a vertical discontinuity caused by an empty slice, causes all remaining regions to be culled. The road is then determined to end (within the image at least) at the top-most remaining region. The result is a series of segments, one per horizontal road slice, that represent where the road is most dense within that slice. Each segment has associated with it a mass (the number of pixels within the segment), the

coordinates of a bounding box and the centre of mass of the segment.

VI. Planning and Plotting a Trajectory

The final step in tracking the road is planning a trajectory that flows safely and smoothly along the road. The centres of mass of the segments described above are used to determine where the trajectory should flow. After an initial "connect-the-dots" approach resulted in a path that was often somewhat rough, it was decided that using weighted centres of mass as control points for spline curves would provide more desirable results. A variation of a simple approximating cubic spline was utilised. Details are presented here for completeness.

The spline is controlled by four control points $p_n = (x_n, y_n, w_n),\ n = -1, 0, 1, 2$. A slice segment's centre of mass determines a point's (x, y) coordinate (with x extending vertically in the image and y extending horizontally) and its mass determines its weight w. A piece-wise curve section between adjacent control points p_0 and p_1 is described by

$$\left.\begin{aligned} y(t) &= \sum_{k=0}^{3} c_{yk} t^k \\ x(t) &= \sum_{k=0}^{1} c_{xk} t^k \end{aligned}\right\} \text{for } 0 \le t \le 1$$

where

$$\begin{aligned} c_{y0} &= y(0) \\ c_{y1} &= y'(0) \\ c_{y2} &= -y'(1) - 2y'(0) + 3y(1) - 3y(0) \\ c_{y3} &= y'(1) + y'(0) - 2y(1) + 2y(0) \\ c_{x0} &= x(0) \\ c_{x1} &= x(1) - x(0) \end{aligned}$$

and

$$\begin{aligned} y(0) &= \frac{y_{-1}w_{-1} + \gamma y_0 w_0 + y_1 w_1}{w_{-1} + \gamma w_0 + w_1} \\ y(1) &= \frac{y_0 w_0 + \gamma y_1 w_1 + y_2 w_2}{w_0 + \gamma w_1 + w_2} \\ y'(0) &= \frac{(y_1 - y_{-1})(x_1 - x_0)}{x_1 - x_{-1}} \\ y'(1) &= \frac{(y_2 - y_0)(x_1 - x_0)}{x_2 - x_0} \\ x(0) &= x_0 \\ x(1) &= x_1 \end{aligned}$$

where γ is a centre-weighting factor with $\gamma = 2$ being used. The use of such a spline results in a trajectory that smoothly approximates the road and that tends toward the most massive (and therefore what is assumed to be the safest) segments of road.

VII. Experimental Results and Discussion

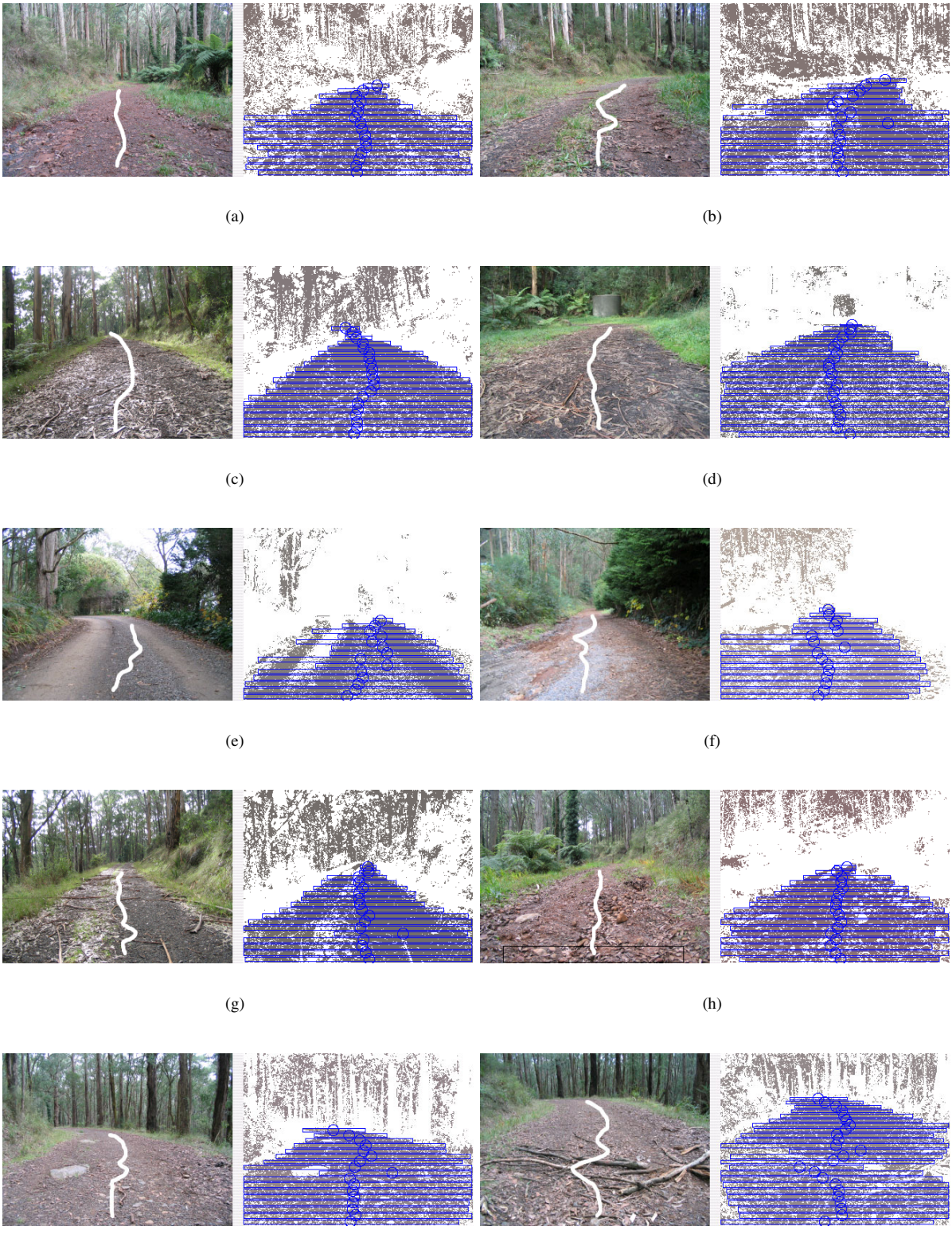
As mentioned, the roads being investigated are poorly structured; in particular they are of variable width and have no markings. As such, a quantitative measure of performance (such as the calculated path's deviation from the road centre) is near-impossible to obtain; qualitative evaluation and verification are more appropriate.

Experimental evaluation consisted of sending a visual feed of a dirt road through the software and observing the generated path. A wide variety of road surfaces were used for testing, with gravel, stones, clay, mud and dirt being most common. Partially obscured road sections were also utilised. For example, areas with heavy leaf-litter, fallen branches/twigs, (shallow) puddles and grass coverage were all evaluated. This was done to ensure the system was tested under *real-world* conditions. The nature of the software (that is, it being a standalone system, not directly linked with a platform or motion control system) meant that a large collection of visual road data could be collected as weather permitted and analysed later.

Figure 4 shows a selection of the roads tested. Each image shows the original input image on the left, overlaid with the spline trajectory. The right side shows the colour-filtered image and road slices (bounding boxes and circles at the centres of mass). Three important things can be seen in these images. Firstly, in most instances the colour filtering is able to remove the majority of the unwanted background while retaining the road surface. Secondly, even when the road appears similar to the background (e.g. 4(a), 4(b), 4(c)) or specific non-road objects (the concrete water-tank in 4(d)) the road slice creation and culling procedure is able to ignore the surrounds and accurately contain the desired road regions. Thirdly, while the trajectories created are not necessarily optimal (nor are they intended to be), they can be seen to follow the centre of the road to a high degree. The use of splines with weighted control points helps to keep the trajectory smooth while pulling it toward "denser" (and hopefully safer) road regions.

The process is also seen to be robust against variations in the road surface and lighting. Mixtures of different surface types are handled comfortably, as shown in 4(b) (dirt and leaf-litter with a grassy centre-strip), 4(c) (dirt and leaf-litter with heavily highlighted areas due to wet leaves and strong, direct sunlight), 4(e) (gravel and tyre-worn dirt), 4(f) (gravel progressing into clay), 4(g) (dirt with sun-highlighted damp grass and leaves), and 4(h) (stones on clay with moderate intensity variation due to the stones). The repeated sampling of the road to obtain new filter parameters makes temporal variations in the road surface (that is, changes between frames) manageable, while the HSI-based parameterisation and filtering allows the system to cope with even relatively large spatial variations (those seen within a single image).

The process of colour filtering and slice creation also has the interesting behaviour of tending to avoid obstacles. Figure 4(i) highlights this behaviour, where an embedded rock causes a sudden shift of the centre of a slice, resulting in



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Fig. 4. Experimental results: roads with trajectories, filtered slices

the trajectory veering away from the potential obstacle. An alternative behaviour exists when generally long and narrow obstacles (such as fallen branches) are encountered. Figure 4(j) illustrates how the created trajectory tends to approach the obstacle perpendicularly, which would result in a more easily traversable path for most platforms.

These two behaviours are the result of a visually distinct obstacle breaking up the fluidity of the road slices. For a simple obstacle like the rock, a void is created in the filtered image where the rock appears and the road slices exist on only one side (horizontally speaking) of the obstacle. More complicated obstacles like the branches create voids that appear on alternate sides of the road centre (first on the right then on the left in 4(j)). These in turn cause the trajectory to swing firstly away from the obstacle as it avoids the first void and then back across the obstacle as it tries to stay away from the second void. These behaviours are clearly useful for a robot intending to traverse potentially hazardous terrain and are an inherent part of the overall road detecting system (that is, they are not the result of additionally included "intelligence" or processing). Further testing is required, however, to determine whether such behaviour can be reliably exploited for the purpose of obstacle avoidance.

Experiments have also shown that under certain circumstances the road detection can be fooled. In particular, if surrounding areas appear similar to the road and are not separated from it distinctly enough, the trajectory can flow from the road and into the background. Because the slice corresponding to the surrounding area must be larger than the conflicting road slice to be chosen over the (correct) road slice, the problem almost only ever appears at the end of a road in the image (where it is furthest and hence appears the narrowest). The end result is a trajectory that initially follows the road but continues into the background beyond where the visible road ends, or forks off the road in the distance and into the background. This effect can be seen to a low degree in Figure 4(c). This is not, however, seen as a serious problem, as the system is intended to run continuously, meaning that (most likely), by the time the problematic trajectory segment is reached, more road will be visible and the trajectory will still follow the road.

A further situation resulting in a temporary breakdown arises when the road surface changes distinctly and rapidly. Under these conditions, the determined road slices stop well short of where the road actually ends and the trajectory similarly ends prematurely. Again, though, this is not seen as seriously detrimental and if anything is potentially useful. If the length of the generated trajectory was used to regulate the robot's speed, such a situation would cause the platform to slow down, causing more parameter updates and a better determination of whether the road simply changed in appearance or actually ends. In the case of a semi-autonomous robot, this could result in a warning being sent to the operator who could then take appropriate action.

VIII. Conclusions

Experimental work has shown the road detecting system to be practically robust and reliable in the face of a wide variety of commonly encountered dirt road surfaces. It successfully differentiates the road from the surrounding areas and is able to plot a smooth trajectory along the road, with the aim of using such a path for motion control. The use of HSI-based colour road parameterisation and filtering allows it to handle many different surface and lighting conditions, including gravel, stones, clay, mud and dirt. Additionally, road cluttering such as that from heavy leaf-litter, fallen branches and grass patches do not cause problems. The method of creating and culling road slices results in visually similar background noise being ignored and the main road area being well contained. A resultant behaviour of the slice creation process is the avoidance of simple obstacles and the creation of a relatively safe path across more complicated ones. Finally, the use of weighted splines for describing the road trajectory results in a smooth flowing path that follows the road centre and can be used for controlling the motion of a robot platform intending to follow the road. Overall, the use of simple but robust algorithms and techniques has resulted in a system capable of separating certain classes of roads from their surrounds and providing a suggested trajectory at a usable processing speed of approximately eight frames per second.

References

[1] D. Fernandez and A. Price, "Visual odometry for an outdoor mobile robot," in *Proceedings of the 2004 IEEE Conference on Robotics, Automation and Mechatronics*, 2004, pp. 816–821.

[2] D. Kuan and U. Sharma, "Model based geometric reasoning for autonomous road following," in *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, vol. 4, Mar 1987, pp. 416– 423. [Online]. Available: http://ieeexplore.ieee.org/iel5/8153/23644/01088049.pdf

[3] R. Lotufo, A. Morgan, E. Dagless, D. Milford, J. Morrissey, and B. Thomas, "Real-time road edge following for mobile robot navigation," *Electronics & Communication Engineering Journal*, vol. 2, no. 1, pp. 35–40, Feb 1990. [Online]. Available: http://ieeexplore.ieee.org/iel1/2219/2620/00080014.pdf

[4] P. Corke, D. Symeonidis, and K. Usher, "Tracking road edges in the panospheric image plane," in *Proceeding of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, Oct 2003, pp. 1330–1335. [Online]. Available: http://ieeexplore.ieee.org/iel5/8832/27959/01248829.pdf

[5] D. Kang, J. Choi, and I. Kweon, "Finding and tracking road lanes using "line-snakes"," in *Proceedings of the 1996 IEEE Intelligent Vehicles Symposium*, Sep 1996, pp. 189–194. [Online]. Available: http://ieeexplore.ieee.org/iel3/4275/12309/00566376.pdf

[6] J. Crisman and C. Thorpe, "Unscarf-a color vision system for the detection of unstructured roads," in *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, vol. 3, Apr 1991, pp. 2496–2501. [Online]. Available: http://ieeexplore.ieee.org/iel2/347/3640/00132000.pdf

[7] D. Raviv and M. Herman, "Visual control signals for road following," in *Proceedings of the 1991 IEEE International Symposium on Intelligent Control*, Aug 1991, pp. 436–442. [Online]. Available: http://ieeexplore.ieee.org/iel2/571/4770/00187397.pdf

[8] 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*, vol. 2, Dec 2002, pp. 625–630. [Online]. Available: http://ieeexplore.ieee.org/iel5/8741/27724/01238496.pdf