Automatic driving on ill-defined roads: an adaptive, shape-constrained, colour-based method

Marek Ososinski **Frédéric Labrosse**[*]
Department of Computer Science, Aberystwtyh University
Aberystwyth, SY23 3DB, United Kingdom

Abstract

Autonomous following of ill-defined roads is an important part of visual navigation systems. This paper presents an adaptive method that uses a statistical model of the colour of the road surface within a trapezoidal shape that approximately corresponds to the projection of the road on the image plane. The method does not perform an explicit segmentation of the images but instead expands the shape sideways until the match between shape and road worsens, simultaneously computing the colour statistics. Results show that the method is capable of reactively following roads, at driving speeds typical of the robots used, in a variety of situations while coping with variable conditions of the road such as surface type, puddles and shadows. We extensively evaluate the proposed method using a large number of datasets with ground truth (that will be made public on our server once the paper is published). We moreover evaluate many colour spaces in the context of road following and find that the colour spaces that separate luminance from colour information perform best, especially if the luminance information is discarded.

1 Introduction

In this paper, we tackle the problem of making a robot automatically drive on a road. The concept of *road* is abstract and hard to define. The term *road* is used for concepts such as a trail, path, route and street, i.e. a surface on which vehicles can travel. We define a *road* as a traversable, linear structure with a colour that is distinguishable from that of its surroundings, that extends for some distance in front of the vehicle and is wide enough for the vehicle. We do not consider the case of roads that are only defined by the fact that they happen to be traversable areas of the ground and not delimited by any change in ground nature, such as desert roads[1]. Moreover, we do not assume anything about the geometry of the road, such as straightness, constant width or well behaved edges.

A road therefore has a surface, an edge and a direction. Any of those properties can be used for road detection and following. The problem with detection lies in the fact that these properties vary between different road types and along given roads. Because of that variability, researchers have proposed the use of neural networks trained to recognise specific road configurations directly from images (Pomerleau, 1992; Jochem et al., 1995) and output driving vectors. However, what is normally the strength of such methods appears to also be their downfall in this particular application: there is just too much variability to either be able to build a representative training set of images or train the neural networks to learn the variability. To solve this

[*] Corresponding author: `ffl@aber.ac.uk`

[1] However, such roads often present a colour that can be slightly different from the surroundings of the road because of being frequently travelled. In such a situation, our method could possibly work, but we have not tested such a situation.

problem, some methods rely on external processes that make the neural network use different parameters based on the type of road being driven (Rosenblum, 2000). Another problem with neural networks using images as direct input is that they generally cannot deal with full resolution images or even full resolution of the colour representation or neural network parameter space (Bao et al., 2012). However, some neural network based method process data that is the result of early image processing to be able to tackle larger images or more complex aspects of images (Jochem and Baluja, 1993; Rosenblum, 2000; Jeong et al., 2002; Chen and Tai, 2010; Shinzato et al., 2011).

Safety aspects of road following have made many researchers opt for more engineered solutions, such as the methods presented next and the proposed method. Early methods extracted features from the road, generally painted markings on structured roads, but, more recently, also the edge of unstructured roads. The second step of such methods is to fit a model of the road to these features. These methods however need either road markings or a well marked edge, which are often missing in ill-defined roads. This is why many more recent methods instead segment the images into road and non-road regions to then build a model used for driving. The literature on road detection and following is very large. We cite here only a subset, trying to show the breadth of the existing methods, in particular in the context of unstructured roads.

Features used to detect the boundary of the roads often are edge pixels, detected using either generic edge detection methods such as gradient (Kluge et al., 1998; Park et al., 2003) or Canny (Wang et al., 2004; Bai et al., 2008) or filters that are tuned based on previous detections (Morgan et al., 1988; Dickmanns and Mysliwetz, 1992; Kluge et al., 1998). Road edges can also be extracted assuming homogeneity of the road colour and/or luminance (Li et al., 1998) or from a distance image computed using a distance measure and colour statistics extracted from the original images (Broggi and Cattani, 2006). Extracted features can also correspond to painted road markings and are detected based on colour properties such as yellow hues (from HSV) or high values of brightness (Kluge and Thorpe, 1995).

Once features have been extracted, a model is fitted to them in order to represent the road. A variety of geometric models have been used to represent the shape of the road. Some are potentially very close to the road, using splines (Wang et al., 2004) or free-form curves (Broggi and Cattani, 2006). Others use a variety of curves such as circles (Morgan et al., 1988; Kluge and Thorpe, 1995), hyperboles (Bai et al., 2008), parabolas (Kluge et al., 1998; Park et al., 2003). More appropriate to road vehicles, clothoidal models have been used in (Dickmanns and Mysliwetz, 1992). Some authors assume that the road is locally straight and use linear models (Paetzold and Franke, 2000). This has the advantage of a simpler model that offers the possibility of modelling roads that are irregular in shape (such as urban roads). Some methods have produced representations in the world coordinate system, these therefore being amenable to (possibly long term) path planning. Others have produced representations in the image plane, these being more appropriate to performing reactive driving.

Methods that either extract road features such as painted lines or explicitly detect the edges of the road are usually not appropriate for ill-defined roads because such features are often missing, not well defined enough to be detectable or confused with other events such as shadows, inhomogeneous cover of the surface or defects such as cracks. Moreover, fitting models to pixels extracted in such a way is error prone because of sparsity of the data and noise in the detection. With the recent challenges such as the DARPA Grand Challenge or the European Landrobot trials (ELROB), emphasis has been moved to methods that do not rely on road markings but instead use the whole surface of the road (and non-road) areas. These methods usually segment the images into road and non-road regions from which models are built.

The precursors of that type of method were probably the SCARF and UNSCARF systems (Crisman and Thorpe, 1988; Crisman and Thorpe, 1991; Crisman and Thorpe, 1993) where segmentation is performed based on Gaussian modelling of the colours of road and non-road pixels using up to twelve Gaussian distributions and the road is modelled as a triangle fitted to the segmented road, the tip being at the vanishing point of the road and the base corresponding to the width of the road close to the robot (bottom of the images).

The results of segmenting images into road and non-road regions depend largely on the information used from the images. In most cases, colour information is used and represented as multiple Gaussian distributions (Jochem and Baluja, 1993; Thrun et al., 2006) or sometimes as histograms (Zhang and Kleeman, 2006; Alvarez and Ĺopez, 2011) or a more complete set of properties such as mean, entropy, variance and energy (Shinzato et al., 2011). Most authors model the colour of the road and non-road areas. When Gaussian distributions are used (the majority of cases) there is a trade-off between the number of distributions used and what each represents of the road and non-road areas (early versions of SCARF started with twelve distributions per class (Crisman and Thorpe, 1988) but only used four in later versions (Crisman and Thorpe, 1993)). This can in particular be a problem when low resolution images are used as each Gaussian will not represent many pixels.

A large variety of colour spaces is used in the literature: RGB (Jochem and Baluja, 1993; Thrun et al., 2006; Procházka, 2013), normalised RGB (Chen and Tai, 2010), HSV (Zhou and Iagnemma, 2010), log-chromaticity and illumination invariance based on camera colour calibration (Alvarez and Ĺopez, 2011), and sometimes a combination of many colour spaces (Shinzato et al., 2011). Few authors use additional information such as texture (Zhou and Iagnemma, 2010). Often the choice is not justified, although a justification for most colour spaces could be given. We are however not aware of a systematic comparison having been made of the different colour spaces in the context of road following.

Usually the nature of roads changes as the vehicle drives, due to changes in illumination and/or surface and localised patterns such as puddles and shadows. This implies that the colour model should be updated as the vehicle travels at a rate that is adapted to the situation. Most methods update the model from previous images, usually using the result of the segmentation and sometimes iteratively segmenting images and updating the model (Crisman and Thorpe, 1993; Zhou and Iagnemma, 2010). In some cases, the colour model from previous images is updated using a model built from specific areas of the images assumed to correspond to road and non-road areas (Chen and Tai, 2010; Lee and Crane III, 2006) or determined as free space using other sensors (Thrun et al., 2006). In a few cases a new model is used for each image from specific areas (Zhang and Kleeman, 2006; Alvarez and Ĺopez, 2011). Updating the colour model seems paramount and not using previous models is dangerous as this could lead to wrong detection of the road should the previous detection be wrong.

Given a model of the road (and non-road) areas, segmentation is performed using a variety of methods, including the use of likelihood measures (Zhang and Kleeman, 2006; Alvarez and Ĺopez, 2011; Procházka, 2013), fuzzy logic (Chen and Tai, 2010), fuzzy support vector machine (Zhou and Iagnemma, 2010), expectation maximisation (Lee and Crane III, 2006), sometimes the result being processed with morphological filters (Zhou and Iagnemma, 2010; Alvarez and Ĺopez, 2011).

Often representations on the image plane are triangular, the tip of the triangle corresponding to the vanishing point of the road (Crisman and Thorpe, 1988; Crisman and Thorpe, 1991; Crisman and Thorpe, 1993). Such a model is usually inappropriate for highly curved and/or non-flat roads as the vanishing point is often not visible in such cases. Some methods have therefore used trapezoids (Jochem and Baluja, 1993; Jeong et al., 2002), sometimes asymmetrical (Jeong et al., 2002) to allow non-straight or non-aligned roads. In a few cases more general, free-form, models are used (Zhou and Iagnemma, 2010), sometimes with geometrical constraints such as symmetry around the road axis (Procházka, 2013). Sometimes there is an explicit assumption of constant (or slowly varying) width (Jochem and Baluja, 1993) and/or straight roads (Zhang and Kleeman, 2006). However, often the same assumption is made but implicitly, specifically when triangular or trapezoidal shapes are used. The non-symmetry of these shapes is only there to help fitting a model when the vehicle is not aligned with the road.

Models are fitted to the segmentation using a variety of methods ranging from Hough transforms (Jochem and Baluja, 1993) to aligning edges or corners of the model to the segmentation (Jeong et al., 2002). Sometimes the central axis of the road is extracted and the model built around it (Procházka, 2013). The fitting of shapes usually results in a best fit and therefore can locally over and/or under estimate the width of the road. This compounded with the constant width assumption can lead to errors in the road following.

Most published work use forward looking cameras, but a few exceptions use panoramic cameras (Zhang and Kleeman, 2006) where properties of the projection are explicitly used in the modelling of road shape, leading to different geometrical models of the road.

Temporal and/or geometrical consistency is often used through algorithms such as Condensation (Isard and Blake, 1998; Bai et al., 2008) or Kalman filters (Paetzold and Franke, 2000; Dickmanns and Mysliwetz, 1992; Jeong et al., 2002; Zhang and Kleeman, 2006; Procházka, 2013). Such an approach has the advantage of speeding up the tracking from image to image but has the drawback of assuming regularity of the roads, and in particular their width, which is often not the case for ill-defined roads.

A number of methods do not fall under the feature or segmentation based categories. For example, the vanishing point is detected in (Rasmussen, 2004) by grouping local orientations on the road. The local orientations are detected using Gabor filters and are used in a voting scheme to find the vanishing point. To tackle the problem of noise created, e.g., by non planar roads, the voting is tracked from the bottom of the images to the top using a particle filter. In (Lieb et al., 2005), reverse optical flow is used to predict the colour properties of the road based on previous experience. The method works well for changes in appearance due to changes in view point but not temporal changes in lighting or colour. In (Luettel et al., 2009), "driving by tentacles" is used. The idea is that feasible trajectories are produced, as a set of tentacles, from the current vehicle configuration. Each tentative tentacle is evaluated based on a number of criteria, one being as low colour saturation as possible, the saturation being computed by converting RGB to HSI. The reasoning is that such driving mechanism is used under low GPS reception, i.e. typically under foliage coverage were indeed colour saturation will be low, and free space will tend to have lower colour saturation than obstacles. This is obviously not the case for more general situations.

Segmentation methods often fail for a number of reasons. One is inadequate representation of the colours. Some authors used several Gaussians to represent the variability of the colours. However, we found that often the various colours present on the road are also present off the road, such as the grass growing on the middle of gravel roads. Explicitly including these colours in the model is therefore likely, depending on the segmentation method used, to wrongly segment parts of the road. Other methods only use one Gaussian and fail to segment the images properly because of local changes in colour due to shadows or wet patches. Most authors also explicitly represent the colours of the non-road areas. This proves to be difficult in most cases due to (1) the variability of these areas and (2) the sometimes similitude between road and non-road colours. Incorrect segmentation then can lead to erroneous fitting of the model.

In contrast, our method uses a shape constrained statistical model of colour information captured from the road at the start of the process. The shape roughly matches that of the projected road onto the image plane while the colour is represented as a single Gaussian (a mean colour and standard deviation of the colour in the shape). The statistical model is updated as the road is followed. We do not perform an explicit segmentation of the road but instead iteratively enlarge the shape to fit the road area using the colour statistics of the road and the Mahalanobis distance. Our fitting method ensures that the model remains inside the road area not including non-road pixels. This implies that when the road width is not constant or the vehicle not aligned with the road the model directs the vehicle in the right direction. We do not model the non-road colours due to their often highly variable nature. This work is an extension of (Ososinski and Labrosse, 2012).

The remainder of the paper is organised as follows. Section 2 presents the various aspects of our method, including model (Section 2.1) and matching to the data (Section 2.2), detection of the road (Section 2.3), adaptation of the model (Section 2.4) and colour spaces used (Section 2.5). Extensive results are given in Section 3 covering aspects such as colour spaces (Section 3.2), effect of adaptability (Section 3.3), repeatability and long-range driving (Section 3.4). The experiments are performed on a large number of datasets that cover a large number of situations (Section 3.1). Section 4 discusses the presented system and results and concludes the paper.

2 Methodology

2.1 Road shape and colour

Our method finds a road-like structure in the current image, based on a statistical model of the colours of the road. Figure 1 shows the projection of a road onto the image plane (we used for this work the front section of images captured by a panoramic camera, see Section 3) along with the isosceles trapezoid used to characterise the road. The parameters of the shape are its width at the top w, position x, legs angle from vertical θ, height h and offset o from the bottom of the image. The road apparent shape depends on the actual road shape and the viewpoint. An increase in road width translates into an increase of w and θ whereas an increase in the height of the viewpoint causes θ to decrease. An offset of the viewpoint from the centre of the road causes the projection of the road to become asymmetrical. The height h of the model depends on the viewpoint height and how far ahead the road needs to be detected. The height h and offset o of the shape are directly related to the smoothness and accuracy of staying within road bounds (Land and Horwood, 1995). The inclusion of the area of the road close to the true horizon increases the smoothness of turning whereas the inclusion of the base of the shape increases the accuracy of staying within road bounds. Offset and height could be modified to fit a given purpose by adjusting the view of the road. In this work, based on the configurations of our robots[2], we use a fixed height $h = 22$ pixels, offset $o = 3$ pixels and $\theta = 42°$ with a symmetrical model. The variable parameters to be determined are therefore w and x.

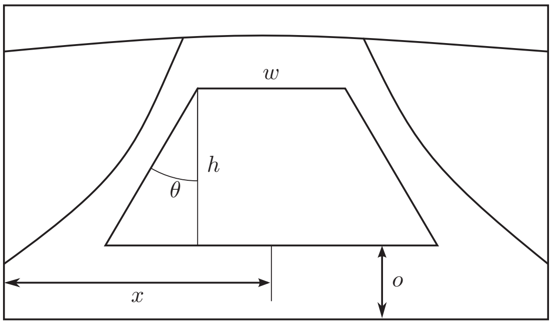


Figure 1: Shape constraint of the model

This trapezoidal shape is only an approximation of the projection of the road surface area onto the images. In a perfect case, the edges of the road would project as curved lines on the image plane and ill-defined irregular and non-flat roads would typically require free-form curves to model them properly. However, the literature shows that more constrained geometrical models are sufficient to represent the traversable area as most methods use the detection of the road only to steer the vehicle. Many authors have used a trapezoidal (or triangular) shape and their results (and ours presented later) show that it is a good enough approximation (Jochem and Baluja, 1993; Crisman and Thorpe, 1993; Zhou and Iagnemma, 2010; Jeong et al., 2002). Such a shape also filters out irregularities of the road, an important property in our context. If an exact fit was sought for, the symmetrical shape would imply that the robot is correctly aligned on the road and that the road is straight. Similarly, the model would assume that the road is locally of constant width. These are assumptions that have been made in other works (see Section 1) and have been shown to be acceptable. Note however that our algorithm is robust against the breaking of these assumptions (see Section 2.3).

The nature (and therefore colour) of the surface of most roads is not spatially constant. To capture this variability, some authors have used a mixture of Gaussians to encode the properties of the road surface (Section 1). Computing such model however is expensive and selecting the correct number of Gaussians is non trivial. Moreover, most local variations of the appearance of the road will either come from shadowing or

[2] The parameters used here are suitable for the narrowest road our robots can drive through (the differences in geometry of our two robots make the shapes similar for both).

wet/dry patches or a central band being of a different nature than the outside bands where wheels often roll. The former can be dealt with using appropriate colour spaces (Section 2.5). The latter usually produces a central band of a similar nature to that of the non-road areas (usually a grassy centre), which should therefore not be part of the road model. Hence we use a single Gaussian to represent the colour of the road. The mean colour and its variance is calculated using the pixels inside the trapezoidal shape that fits the road. It is assumed that the various colour components are independent (Section 2.5).

2.2 Road matching

Once a colour-based Gaussian statistical model of the road is computed, the similarity between the model and each pixel's colour can be evaluated. The more similar the colours are, the more likely the corresponding pixel is to belong to the road. The squared Mahalanobis distance is used to this effect:

$$M(p) = \sum_{i=1}^{C} \frac{(\mu_i - p_i)^2}{\sigma_i^2}, \tag{1}$$

where p is the colour of a pixel, C is the number of colour components, the subscript i indicates the i^{th} colour component, μ is the mean colour and σ is the colour standard deviation of the model. This is similar to the method used in (Thrun et al., 2006), except that we only use a single Gaussian.

If we define $\mathcal{P}(w, x)$ as the set of pixels belonging to the shape of parameters w and x (Figure 1), then the distance between that shape on the current image and the road model is given by

$$d(w, x) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}(w,x)} M(p). \tag{2}$$

To be able to detect the centre of the road, the widest possible shape fitting the projection of the road onto the image must be detected. The parameters of the shape best matching the image and the road model are obtained by minimising the error function

$$e(w, x) = d(w, x) + \alpha \frac{1}{w}, \tag{3}$$

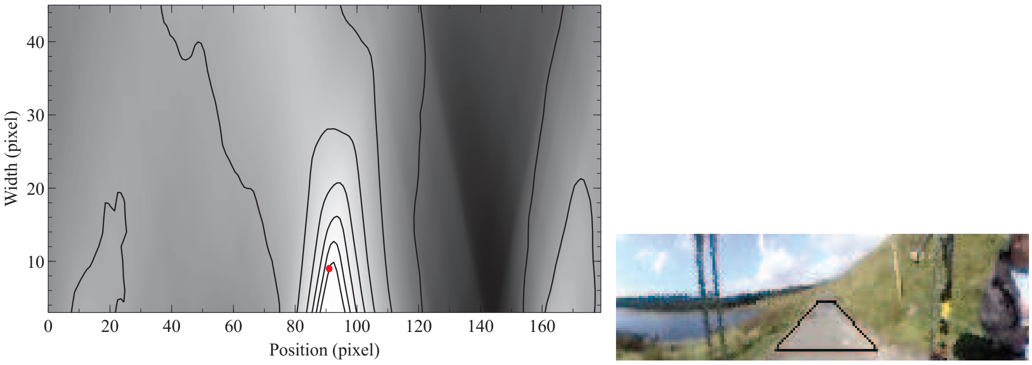
where α is a weight that adjusts the importance of the two terms. This function is smooth (see Figure 2) because the Mahalanobis term is smooth due to the smoothing effect adding or removing entire lines of pixels in $\mathcal{P}$ has (changing either parameter adds and/or removes lines of pixels at the edges of the trapezoid). Moreover, the area around the minimum is flat for shapes that are too narrow (the shape matches equally well for a wide range of positions). Therefore standard minimisation techniques cannot be used as they would provide one of many possible results, but not necessarily the widest road possible. Minimising the function is instead achieved with an algorithm (described in Section 2.3) that finds the shape with the maximum width on that flat minimum.

Note that when Gaussian distributions are compared, the Mahalanobis distance between the distributions can be used (as in (Thrun et al., 2006)). We instead chose to compare individual pixels to the model distribution in (1) and (2) because this allows the incremental comparison of pixels to the model rather than needing to collect statistics first, which would imply multiple evaluations of the distance function.

Minimising (3) is done using two different methods, depending on whether it is the first detection of the road (initialisation step) or while the robot moves. Both methods incrementally compute the error function by starting from a narrow shape positioned in (or near) the flat area and increasing the width of the shape.

2.3 Initialisation of the model and road tracking

The initial detection is performed by estimating the colour parameters of the model of the smallest traversable area (based on the width of the robot and height of the camera, $w = 3$ pixels in all our experiments) in front of the robot and expanding that model based on (3). This procedure assumes that the robot is initially roughly positioned at the centre of the road and roughly aligned with it, using an initial position of the road x corresponding to the front of the robot (see Section 3). Initially, an empirically identified value of $\alpha = 35$ is always used as, according to our experimentation, this is a good compromise to give enough importance to the Mahalanobis term in the error function. From the front position and minimum width, the shape width is (symmetrically) increased until the error value increases (this procedure works due to the smoothness and flat minimum of the error function, Section 2.2). This can result in an underestimation of the road width if the robot is not properly centred on the road or aligned with it, which is not undesirable. Figure 2 shows the error function (3) for the first frame of the LAKESIDE dataset (Section 3.1). The dot shows the result of the initial detection.



(a) Error function. The dot shows the result of initial detection.

(b) Initial detection

Figure 2: Error function and initial detection for the first frame of the LAKESIDE dataset

In subsequent road detections, we use half of the initial shape width as the weight α. Our informal experiments showed that this value provides good results but also that the results do not depend dramatically on the actual value.

After initialisation, subsequent road detections are performed in five stages (Figure 3) while the robot moves:

Stage 1: a zero width model is positioned on the new image at the position of the detection in the previous image;

Stage 2: the model is expanded symmetrically sideways (4 pixels at a time) until the error function (3) increases. The result of this stage is used as the starting point for Stages 3 and 4;

Stage 3: the model is expanded leftwards (one pixel at a time) until the error increases;

Stage 4: the model is expanded rightwards (one pixel at a time) until the error increases;

Stage 5: the position and width of the shape are computed from the left and right positions obtained in Stages 3 and 4.

The role of Stage 2 is to speed up the process by providing a rough estimate of the shape width while Stages 3 and 4 provide a higher resolution estimate. The expansion at Stages 2 to 4 stops at the edge of the road because of the smoothness of the function.

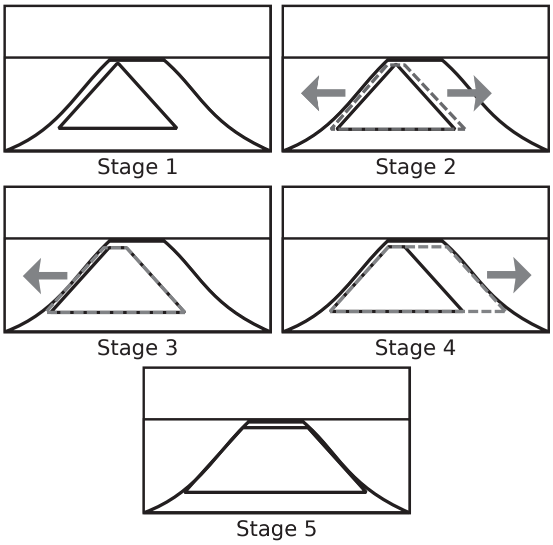


Figure 3: The five stages of detection

The procedure assumes that the rotation of the robot and/or turning of the road is such that the narrow shape at Stage 1 still lies within the road when positioned on the new image where it was detected in the previous image. If this is not the case, then Stage 1 can lead to the shape being positioned at least partially outside the road. If it is not completely outside the road, then the shape will not expand further outside the road during Stages 2 to 4 because any non-road pixel added to the error function will increase its value. If the shape is totally positioned outside the road, then the road can be lost and the system may not recover if this lasts for too many frames. The best way to avoid this happening is to ensure that the processing frame rate is high enough relative to the speed of the robot.

The horizontal expansion of the model ensures that only the road is included in the model. This is particularly useful when the constant width and straight road assumptions are broken because then as only the road (drivable area) is included in the model, the robot is lead away from encroaching road edges or towards the right turn. Moreover this ensures that the updating of the colour statistics based on the colour content of the detection will be less likely to drift towards wrong colours.

2.4 Adaptability

The road surface varies due to changes in viewpoint and environmental factors. The changes could render the model inadequate over time were it not updated, as most authors have noticed. A simplistic approach would be to replace the mean and variance of the colour of the road with that of the newly detected road. However, in the unlikely case of wrong detection of the road, the resulting model could become corrupted, which is especially true if only one Gaussian is used. To avoid this, the road model is updated so that the colour statistics are changed towards the new statistics from the current ones. This offers robustness to wrong detections as well as progressive update of the model.

Moreover, to offer more robustness against including the edges of the road in the shape, only a proportion of the width of the road is used when updating the model, i.e. by using $w_s = \gamma \times w$ for the width of the narrow shape. We used $\gamma = 0.8$ in all our experiments. This value offers a trade-off between ignoring enough of the sides of the road in case of imprecise detection and having enough pixels for narrow roads to calculate colour statistics. For better defined roads, a higher value could be used.

The C colour components of the two model parameters (mean μ_i and variance σ_i^2, $i = 1 \cdots C$) are updated from their current value towards the equivalent value of the newly detected narrow shape (mean and variance of the colours of the pixels in the shape), as defined above. The parameters are updated using

$$
{}^{t+1}u = \begin{cases} {}^{t}u + \varphi v & \text{if } u_s > {}^{t}u, \\ {}^{t}u - \varphi v & \text{if } u_s < {}^{t}u, \\ {}^{t}u & \text{if } u_s = {}^{t}u, \end{cases} \tag{4}
$$

where ${}^{t}u$ is the road model parameter to update (μ or σ^2) at time t, u_s is the corresponding model parameter for the newly detected shape and v a measure of distance between the two parameters ${}^{t}u$ and u_s. This rule is used for each colour component independently. The adaptability factor φ controls how much the new parameter values influence the model parameters. The effect of this parameter is tested in Section 3.3. A good compromise established during testing was found to be $\varphi = 0.05$, although the performance is not dramatically affected by the value as results presented in Section 3.3.1 show.

To update the mean colour, the distance v_μ is taken as the Mahalanobis distance between the current model and the mean colour μ_s of the narrow shape (see (1))

$$
v_\mu = \sqrt{M(\mu_s)}. \tag{5}
$$

Each colour component of the mean is updated using (4). The Mahalanobis distance is used here to take into account the width of the colour distribution of the road model.

Similarly, the distance v_{σ^2} is taken as the Euclidean distance between the current variance and the variance of the colour σ_s^2 of the narrow shape

$$
v_{\sigma^2} = \sqrt{\sum_{i=1}^{C} (\sigma_i^2 - \sigma_{s\,i}^2)^2}. \tag{6}
$$

Again, each colour component of the variance is updated using (4).

In (Thrun et al., 2006) a similar method is used to update a Gaussian towards a newly evaluated one judged to be similar (Mahalanobis distance below a set threshold). However, in their case the new mean and standard deviations are taken as the weighted mean of the corresponding parameters of the two original distributions, where the weighting is a function of the number of pixels each distribution claims. In our case, all pixels are claimed because we use a single Gaussian so instead we move the model towards the new distribution as a function of their distance, controlling the amount of update.

2.5 Colour spaces

The proposed method relies on the assumption that the colour of the road is different from that of the non-road areas of the images. However, such difference depends on how colours are being represented, especially when environmental conditions can change. As can be seen in Section 1, a large variety of colour spaces have been used, often chosen arbitrarily or with some theoretical justification. We are however not aware of a systematic comparison of colour spaces in the context of road following. We therefore perform here such comparison of a variety of colour spaces, some being widely used, others being more exotic. The reliance of our method on colour differences means that it will indeed provide such an objective comparison of the performance achieved with the different colour spaces.

We provide here a detailed description of the conversions used. This section is to be used as a reference section, but is not necessary for the comprehension of the paper. However, note the following convention used throughout the paper: a '_' character in the name of a colour space (as in "HS_") means that the corresponding component is discarded.

The RGB (Red Green Blue) colour space is an obvious choice as this is what most cameras provide. The HSV (Hue Saturation Value) and YUV spaces are traditionally used when "natural" colour specifications are needed, and indeed used in the road following literature. The V component in HSV and Y component in YUV encode the lightness of the colour and discarding these can help dealing with shadows. We therefore also use HS_ and _UV as colour spaces. The transformation from RGB to YUV is the traditional:

$$\begin{aligned}\mathrm{Y} &= 0.299\mathrm{R} + 0.587\mathrm{G} + 0.114\mathrm{B},\\ \mathrm{U} &= 0.492(\mathrm{B} - \mathrm{Y}),\\ \mathrm{V} &= 0.877(\mathrm{R} - \mathrm{Y}),\end{aligned} \tag{7}$$

where $0 \leq \mathrm{R}, \mathrm{G}, \mathrm{B} \leq 1$, $0 \leq \mathrm{Y} \leq 1$, $-0.492 \leq \mathrm{U} \leq 0.492$ and $-0.877 \leq \mathrm{V} \leq 0.877$. The transformation from RGB to HSV is as follows. For $0 \leq \mathrm{R}, \mathrm{G}, \mathrm{B} \leq 1$

$$\begin{aligned}M &= \max(R, G, B),\\ m &= \min(R, G, B),\\ C &= M - m.\end{aligned} \tag{8}$$

With this, HSV is defined as

$$\mathrm{V} = M, \tag{9}$$

$$\mathrm{H} = 60^\circ \times \begin{cases} \text{undefined} & \text{if } C = 0,\\ \frac{G-B}{C} \mod 6 & \text{if } M = R,\\ \frac{B-R}{C} + 2 & \text{if } M = G,\\ \frac{R-G}{C} + 4 & \text{if } M = B, \end{cases} \tag{10}$$

$$\mathrm{S} = \begin{cases} 0 & \text{if } C = 0,\\ \frac{C}{V} & \text{otherwise.} \end{cases} \tag{11}$$

YCbCr is another colour space that separates luminance information from colour information. In this work we consider both keeping the luminance information (using the whole YCbCr space) and only retaining the colour information (using the _CbCr space). The transformation from RGB is given by

$$\mathrm{Y} = 0.299\mathrm{R} + 0.587\mathrm{G} + 0.114\mathrm{B}, \tag{12}$$

$$\mathrm{Cb} = 0.5 - 0.169\mathrm{R} - 0.331\mathrm{G} + 0.500\mathrm{B}, \tag{13}$$

$$\mathrm{Cr} = 0.5 + 0.500\mathrm{R} - 0.419\mathrm{G} - 0.081\mathrm{B}, \tag{14}$$

where $0 \leq \mathrm{R}, \mathrm{G}, \mathrm{B} \leq 1$, $0 \leq \mathrm{Y}, \mathrm{Cb}, \mathrm{Cr} \leq 1$.

A fast and easy way to offer robustness to changes in illumination (and shadows to some extent) is to use the *CIE L*a*b** (labelled 'Lab' in figures) colour space and ignore the luminance component $L*$ (Woodland and Labrosse, 2005). The $a*$ channel specifies the colour as green to magenta hues, whereas the $b*$ channel specifies the colour as blue to yellow hues. We will therefore consider the $\_a*b*$ (labelled '_ab' in figures) colour space (only using colour information and discarding luminance). The transformation from RGB to *CIE L*a*b** is given by

$$\begin{pmatrix} X\\ Y\\ Z \end{pmatrix} = \begin{pmatrix} 2.7690 & 1.7518 & 1.1300\\ 1.0000 & 4.5907 & 0.0601\\ 0.0000 & 0.0565 & 5.5943 \end{pmatrix} \begin{pmatrix} \mathrm{R}\\ \mathrm{G}\\ \mathrm{B} \end{pmatrix} \tag{15}$$

$$L^* = 116 \left(\frac{Y}{Y_n}\right)^{1/3} - 16 \tag{16}$$

$$a^* = 500 \left[\left(\frac{X}{X_n}\right)^{1/3} - \left(\frac{Y}{Y_n}\right)^{1/3}\right] \tag{17}$$

$$b^* = 200 \left[\left(\frac{Y}{Y_n}\right)^{1/3} - \left(\frac{Z}{Z_n}\right)^{1/3}\right], \tag{18}$$

where X_n, Y_n, and Z_n are the X, Y, and Z values of the reference white, $\mathrm{R} = \mathrm{G} = \mathrm{B} = 255$.

In (Katramados et al., 2009), a transform named by the authors 'mean chroma' is used to perform a colour-based segmentation of traversable areas. It is a one-dimensional colour space defined as

$$\mathrm{MCh} = \frac{\mathrm{Cb} + \mathrm{Cr} + 2a^*}{4}, \tag{19}$$

that is a combination of some of the colour information from the YCbCr and *CIE L*a*b** colour spaces. Because that colour transform is rather ad-hoc, we have also created variations based on it. The first was created to check if combining the components in one was making a difference. We therefore evaluated the space CbCra^* (labelled 'CbCra' in figures) that uses the same components as the original MCh but keeps them separate, therefore producing a three-dimensional colour space. As the results show, the performance increases. We have also tried to use the value a^* directly as the third component (instead of $2a^*$); this makes no difference to the results. MCh$'$ is also a linear combination of components, producing a single component in $[0; 255]$, but this time using normalised values of Cb, Cr, a^* and b^*. In this transform we retained the b^* component because we found that this component also contains information useful for the discrimination of road pixels from non-road pixels on some of our datasets. Normalisation of the a^* and b^* components was based on the extreme values of both components across all the images of our datasets. All four components are normalised in the 0 to 255 range:

$$\begin{aligned}
\mathrm{Cb}_n &= \mathrm{Cb} \times 255, & (20)\\
\mathrm{Cr}_n &= \mathrm{Cr} \times 255, & (21)\\
a_n &= (a^* + 99.6749) \times 1.232539626, & (22)\\
b_n &= (b^* + 92.5584) \times 2.433977176, & (23)
\end{aligned}$$

followed by appropriate clamping. Because of non-homogeneity of the coverage of the 0 to 255 range, we define MCh$'$ as

$$\mathrm{MCh}' = \left(\frac{\mathrm{Cb}_n + \mathrm{Cr}_n + a_n + b_n}{4} - 90\right) \times 2.65625, \tag{24}$$

again followed by appropriate clamping. This ensures that, on our datasets, the full range of possible values is used.

The log-chromaticity space (LCS) is often used, e.g. in (Alvarez and Ĺopez, 2011), as the basis for a shadow-free colour transform. This is a two dimensional colour space, defined as:

$$\mathrm{LCS}_1 = \begin{cases} log(r/g) & \text{if } g > 0, \\ log(r) & \text{otherwise,} \end{cases} \tag{25}$$

$$\mathrm{LCS}_2 = \begin{cases} log(b/g) & \text{if } g > 0, \\ log(b) & \text{otherwise.} \end{cases} \tag{26}$$

For the rare cases of r or b being 0, we use 1 instead for their value, therefore still providing a similar value and avoiding the numerical issue.

2.6 Numerical stability

If the model is estimated from a road for which the colour is perfectly homogeneous, the standard deviation vanishes, making the update of the mean colour (5) diverge. In practice, noise in the acquisition implies that the colours are never homogeneous and the standard deviation therefore does not vanish. It can however become too small to be represented using the normal limited precision of floating point numbers when the road is "homogeneous" (parts of some of our datasets RunningTrack and Footpath) and for colour spaces that are by design normalised, i.e. their components are restricted to the $[0; 1]$ interval, which forced standard deviations to be lower than 1.

In these circumstances, updating the model of the road using (4) – (6) would lead to the standard deviation correctly becoming increasingly small (the standard deviation of the model converging towards the measured standard deviation), eventually resulting in an infinite mean colour and then producing invalid Mahalanobis distances between the model and the image pixels. This numerical instability was solved by multiplying all colour components described in Section 2.5 by 100. This does not alter the Mahalanobis distance values (because the correction is linear) but implies larger standard deviation values compared to the original values, therefore resolving the numerical instability. Other transformations would have been possible but non-linear ones would correspond to changing the colour transformations and/or metric used to compare colours.

3 Results

We present in this section results of our method on a number of datasets. The first experiment assesses the viability of the various colour spaces presented in Section 2.5. The second experiment shows the effect of the adaptability factor (Equation (4)) on the performance of the system in detecting the road. Finally, we show repeatability and long-range driving in one of our test areas.

The first two experiments (effect of colour space and adaptability) are performed on pre-captured datasets, described in the next section. Initial tests were conducted with a Pioneer 3AT, with which some of the datasets were captured. The longer driving experiments were conducted with Idris (Figure 4), our four wheel drive and four wheel steering robot, based on a robuCar-TT chassis. This robot is capable of driving on rough surfaces and is therefore ideal for some of the experiments we conducted. The cameras used are catadioptric (hyperbolic mirror) omni-directional cameras and the images are unwrapped into panoramic images (Labrosse, 2006) with one pixel per degree of rotation around the camera. The proposed method does not use the fact that the images are panoramic and can work for forward looking cameras by limiting the position and/or width of the road to the image plane. Because of the type of camera used here, we do not limit the position or width of the road and instead wrap around the images. This allows worse results to be obtained (constraining the detection to the front, where the road is, can only help) but allows us to consider more failure cases. In most cases, only the front part of the images is shown and the road is expected to be at the centre of the images. In a few cases, full panoramas are shown, in which case the road is expected to be at column index 25% of the width of the panoramic images from the left, the back of the robot being at column index 75%, the right of the robot at column index 50% and the left at column index 0% (left-most column).

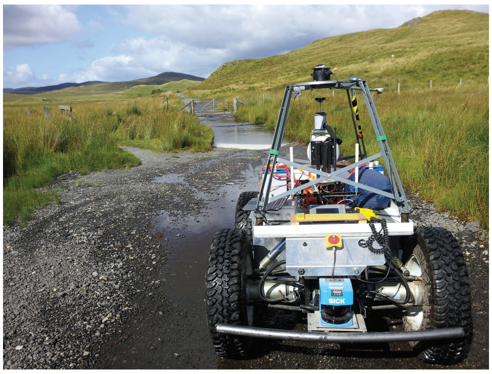


Figure 4: Idris at Point 3 in Figure 26

All detection (position and width) errors and statistics are evaluated against ground truth, which was created by manually annotating the images, “drawing” the trapezoid on each of the images of each sequence using an in-house software package that allows the specification of the trapezoid by two mouse clicks on either side of the trapezoid.

The experiments were performed on unwrapped panoramic images of 360 × 55 pixels. For the off-line experiments, on one core of a 2.7 GHz Intel® Core$^{\mathrm{TM}}$ i5 CPU with local solid state storage, the processing runs at approximately 370 Hz without any logging of data. On-line processing speeds (while logging all data including images, running as part of a complex system in charge of the safety of the robots, and including image acquisition) are given in Table 1 (Pentium® 4M 1.8 MHz on Idris, one core of a dual-core 2.26 GHz Core$^{\mathrm{TM}}$ 2 Duo on the Pioneer).

Only the steering is controlled by the system for the actual driving. The road position is expected to be at a given value that depends on the geometry of the system (column index 91 in the images captured by our robots). The difference between actual and desired position is used in a proportional controller which sets a rotational speed (for Pioneer robots) or steering (for Idris). The proportional constant was the same for both robots and only adequately tuned (i.e. producing a system that follows the road at the typical speeds of the robots).

The speed was manually set to control the effect of adaptability (see Sections 2.4 and 3.3). In a real system, the speed could be linked to the road position error as well as the detected width of the road.

3.1 Datasets

The datasets used in the off-line experiments were captured during road following trials (apart from one) and are described below. They all represent different challenges and are limited in length not by failure of the method but by either end of the road (physical end, entering in an area that is not road-like any more, obstacle, etc.) or lack of further features. For each dataset, we show a number of typical and interesting frames. Table 1 gives the speed of the robot and the frame rate of the capture of the various datasets. The speeds were desired speeds rather than actual speeds, which vary, in particular in the presence of obstacles. The various speeds were chosen based on versions of the software (lower speeds for earlier versions) and specific requirements (Lakeside was intentionally done very slowly). The capture rate is an average and varies by about 1 fps depending on hard drive caching.

Table 1: Speed of robot and Frame Per Second of the image capture for the various datasets

Dataset	Speed (m/s)	Capture (fps)	Robot
RunningTrack	0.2	5	Pioneer
Llanbadarn	1.0	3	Idris
Shadows	1.0	3	Idris
Footpath	0.2	4	Pioneer
WindFarm	0.3	3	Idris
Lakeside	0.1	3	Idris

RunningTrack The running track (Figure 5) is a highly visible manufactured surface accompanied by well defined edge markings. This blue surface contrasts well against the white edges and central line as well as the grassy surroundings. However, the two adjacent running tracks are both some variation of blue, the robot driving on the lighter of the two tracks. This dataset comprises 1803 images. The challenges of the dataset include changes in surface colour caused by leaves, cast shadows by nearby trees, occasional crossing white and yellow lines and an intersection.

Llanbadarn The road (Figure 6) in this dataset is a well marked tarmac surface which contrasts against the mostly green surroundings. This dataset comprises 541 images. The challenges of this dataset include noise (a moving diagonal pattern in the original omni-directional images resulting in hyperbolic lines forming on the unwrapped panoramic images due to noise in the camera), general lack of colour contrast (due to an overcast sky), raised pedestrian crossings, a T-junction and a sharp turn involving a brief change in the road surface.

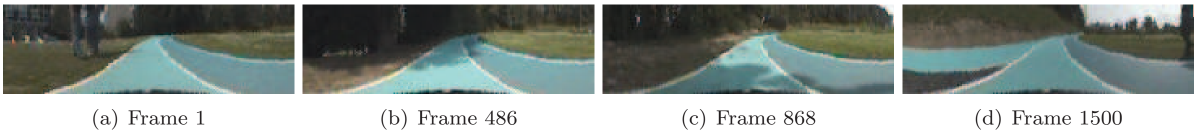
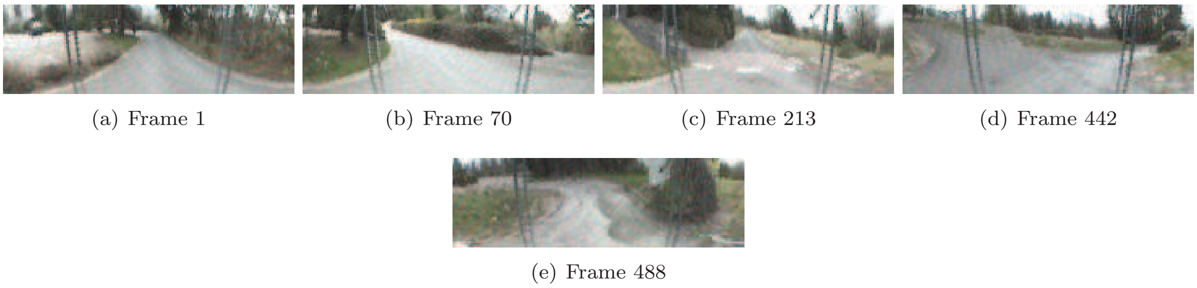

(a) Frame 1 (b) Frame 486 (c) Frame 868 (d) Frame 1500

Figure 5: Frames from the RunningTrack dataset


(a) Frame 1 (b) Frame 70 (c) Frame 213 (d) Frame 442

(e) Frame 488

Figure 6: Frames from the Llanbadarn dataset

Shadows The road in this dataset (Figures 7) is a well marked tarmac surface which contrasts against the mostly green surroundings. This dataset comprises 150 images and is geographically a subset of the Llanbadarn dataset, not including any of the intersections and driven in the opposite direction. The challenges of this dataset involve saturation and therefore high and low (colour) contrasts, coupled with multiple shadow configurations on the road. Note that on that dataset, the ground truth covers the whole road width, therefore including both bright (almost saturated) and shaded areas, while the initial model only contained the bright area (because of initial positioning of the robot).

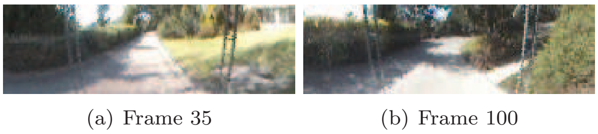
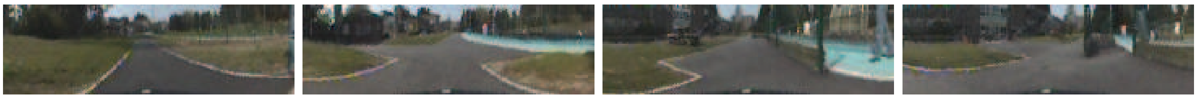

(a) Frame 35 (b) Frame 100

Figure 7: Frames from the Shadows dataset

Footpath The road in this dataset (Figures 8) is a marked tarmac surface which contrasts well against the mostly green (grass) and blue (running track) surroundings. This dataset comprises 1551 images. The challenges of this dataset include a crossroad, a widening of the road and an obstacle. Also, the tarmac is covered in moss on the left hand side at the beginning of the dataset.

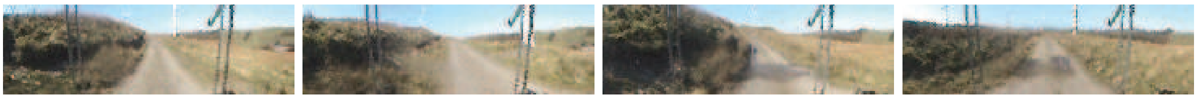
WindFarm The road in this dataset (Figures 9) is an unmarked loose surface which is highly similar in colour to the surroundings. This dataset comprises 2998 images. The challenges of this dataset include low contrast between the road and the surroundings with no crisp edge, surface discolouration due to wet patches and changing viewpoint due to changes in the on-road position. Note also that due to a bug in an early version of the software, the robot had a tendency to “hug” the left side of the road, implying that in most images the road does not fit well the shape used to model it. Moreover, the safety systems of the robot were now and then stopping the robot because it was too close to the tall grass on the left. This resulted in the robot stopping completely for around 102 frames, which were removed, hence a gap between frames 423 and 525 in the presented results.

Lakeside The road in this dataset (Figure 10) is made of various materials ranging from lose grey gravel to brown mud and presents dry and wet patches with puddles in places (see Figure 4 for a typical



(a) Frame 1 (b) Frame 466 (c) Frame 1300 (d) Frame 1400

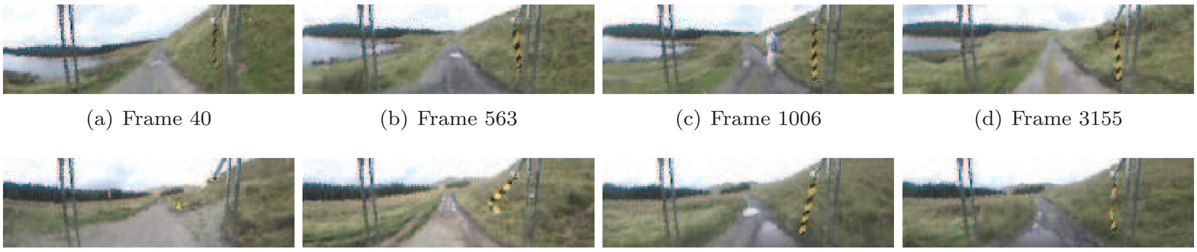
Figure 8: Frames from the FOOTPATH dataset



(a) Frame 80 (b) Frame 730 (c) Frame 1264 (d) Frame 2754

Figure 9: Frames from the WINDFARM dataset

view, the dataset was captured between Points 1 and 2 in Figure 26). The road is delimited by grass but the boundary road-grass is not always obvious. This dataset comprises 11226 images and was captured using teleoperation at low speed (0.1 m/s). The speed was such that decimating the dataset could simulate variable robot and/or processing speeds (Section 3.3). The challenges of this dataset include the changes in appearance of the road, a pedestrian walking in front of the robot (around frame 1006), changes in attitude of the robot due to bumps and holes in the road, occasional central "band" of different colour (grass or rocks) and some intersections.



(a) Frame 40 (b) Frame 563 (c) Frame 1006 (d) Frame 3155

(e) Frame 8350 (f) Frame 8858 (g) Frame 10800 (h) Frame 11025

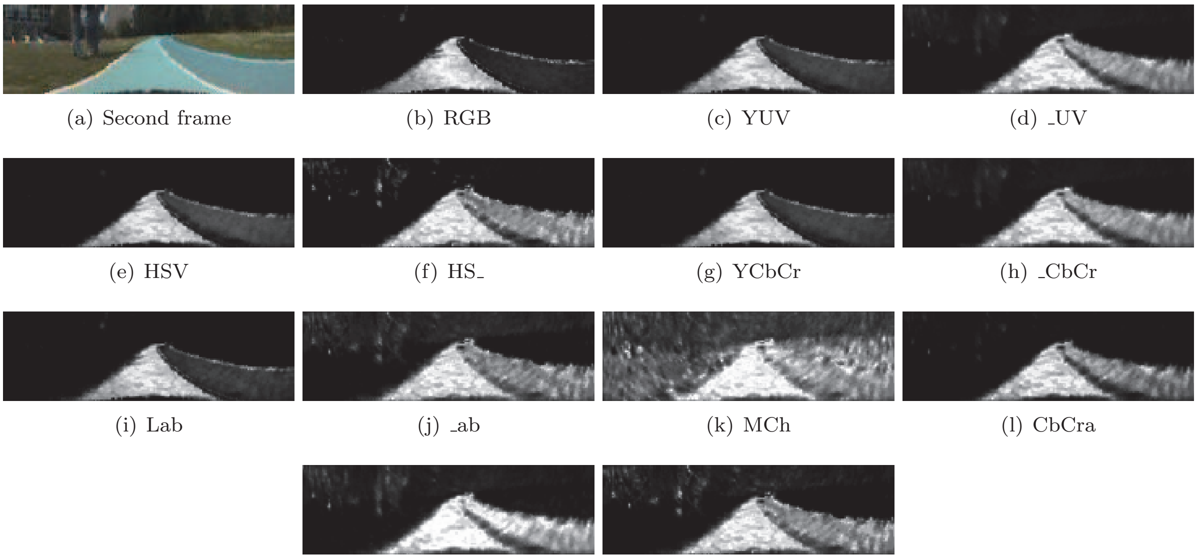
Figure 10: Frames from the LAKESIDE dataset

3.2 Colour transforms

The performance of the colour transforms is mainly related to the accuracy of pixel classification (note however that the proposed method does not perform an explicit classification of the pixels). This is presented as the Mahalanobis distance between pixels of the second frame of each sequence and the initial model derived from the first frame. Comparing the second frame to the first frame ensures that the comparison is fair, removing effects due to changes in road appearance. The results are presented using a logarithmic grey scale so that low Mahalanobis distance values appear light. The logarithmic scale emphasises the differences between low distance values. A good result is therefore one for which the road and only the road is light, or where at least the road is well separated from its surroundings.

The second aspect of these results is to compare the detection (position and width) of the road to ground truth using the various colour spaces. For each dataset we present:

- plots showing the error between detected position and width and ground truth for a selection of colour spaces. Because the successes are similar, which is the case for most colour spaces, only RGB



(a) Second frame
(b) RGB
(c) YUV
(d) _UV
(e) HSV
(f) HS_
(g) YCbCr
(h) _CbCr
(i) Lab
(j) _ab
(k) MCh
(l) CbCra
(m) MCh′
(n) LCS

Figure 11: RunningTrack: Mahalanobis distance of the pixels of the second frame to the initial model derived from the first frame

as the de facto colour space, _ab (for a reason that becomes clear later) and a few specific failures are shown;

- the mean of the position and width errors along with their standard deviation. These are given for all datasets and colour spaces in Tables 2 and 3.

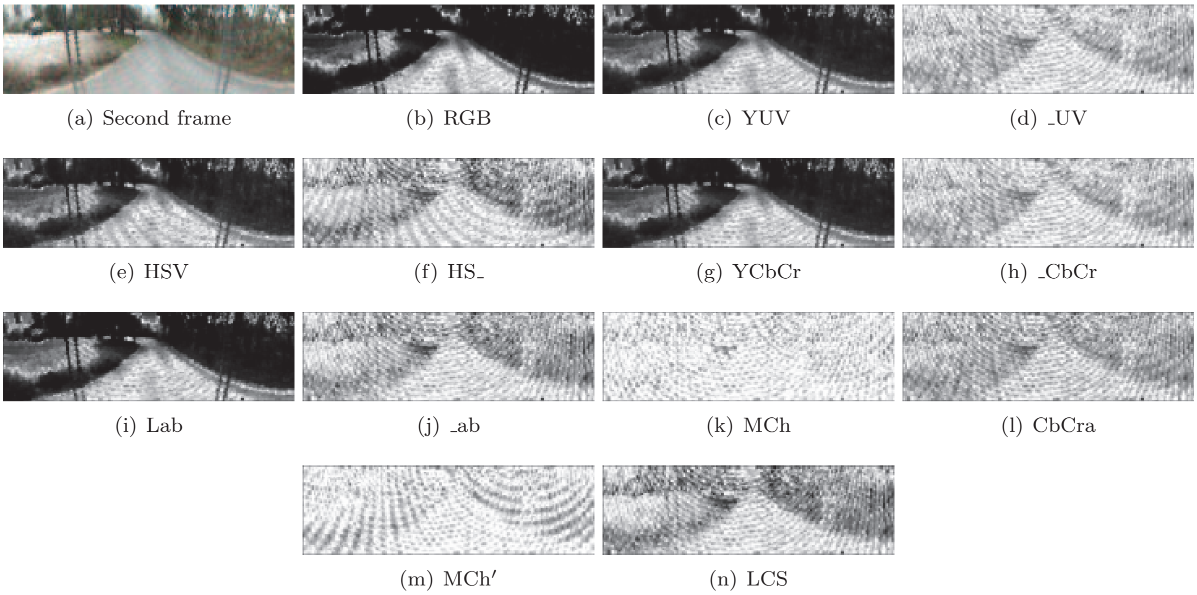
Positive position errors indicate that the road is detected too far to the left while positive width errors indicate an underestimation of the road width. All results presented in this section were computed with an adaptability factor $\varphi = 0.05$ (Equation 4).

3.2.1 Mahalanobis distance images

Visual inspection of the Mahalanobis distance per pixel between the second frame and the road model built from the first frame reveals that some of the colour transforms will perform better than others (Figures 11 to 16). Indeed, some will show a clear low distance value (light pixels in the figures) for the road well contrasted from the high (darker) values surrounding the road.

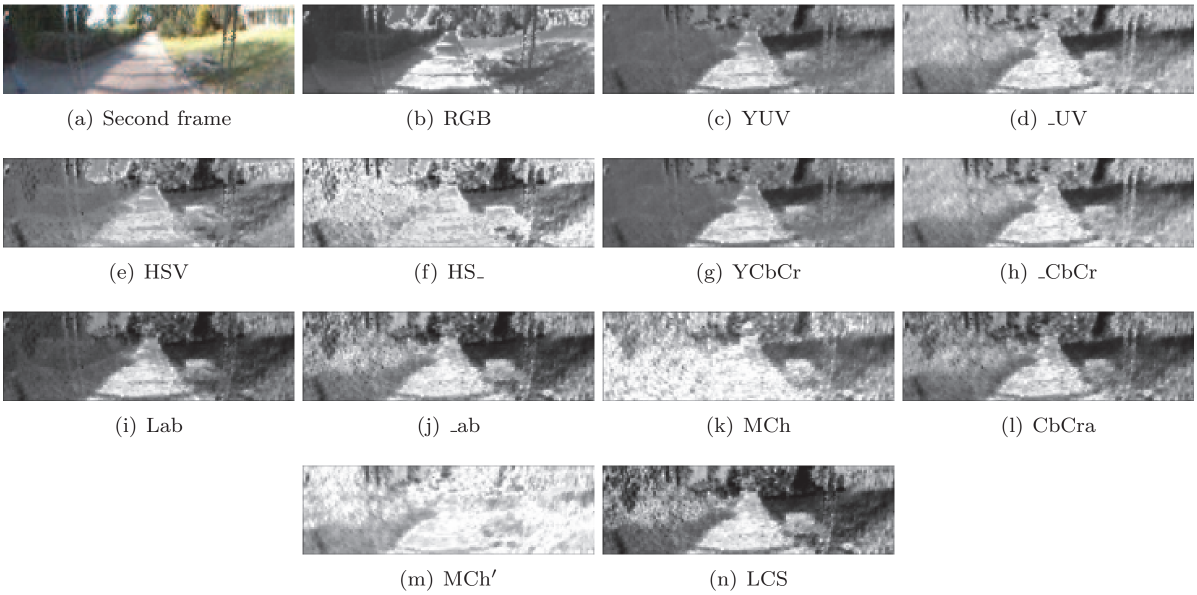
In particular, examining the RunningTrack dataset (Figure 11), RGB seems to be the colour space that separates best the two shades of blue of the running track. However, along with the other colour spaces that retain luminance information, the correct (light blue) path is not so well homogeneously defined and its detection is susceptible to shadows (cast by the authors on the running track) and discolouration (bottom left part of the track).

Generally, colour transforms that retain the luminance information are better at distinguishing between road and non-road pixels (e.g. RGB on the RunningTrack (Figure 11) and Lab compared to _ab on Llanbadarn (Figure 12) or Lakeside (Figure 16)) but are at the same time heavily influenced by the presence of shadows and wet patches. Colour spaces that separate luminance from colour information seem to perform better against shadows than colour spaces that do not, even if retaining the luminance information (compare for example Lab or YUV against RGB in Shadows, Figure 13).



(a) Second frame (b) RGB (c) YUV (d) _UV (e) HSV (f) HS_ (g) YCbCr (h) _CbCr (i) Lab (j) _ab (k) MCh (l) CbCra (m) MCh′ (n) LCS

Figure 12: Llanbadarn: Mahalanobis distance of the pixels of the second frame to the initial model derived from the first frame

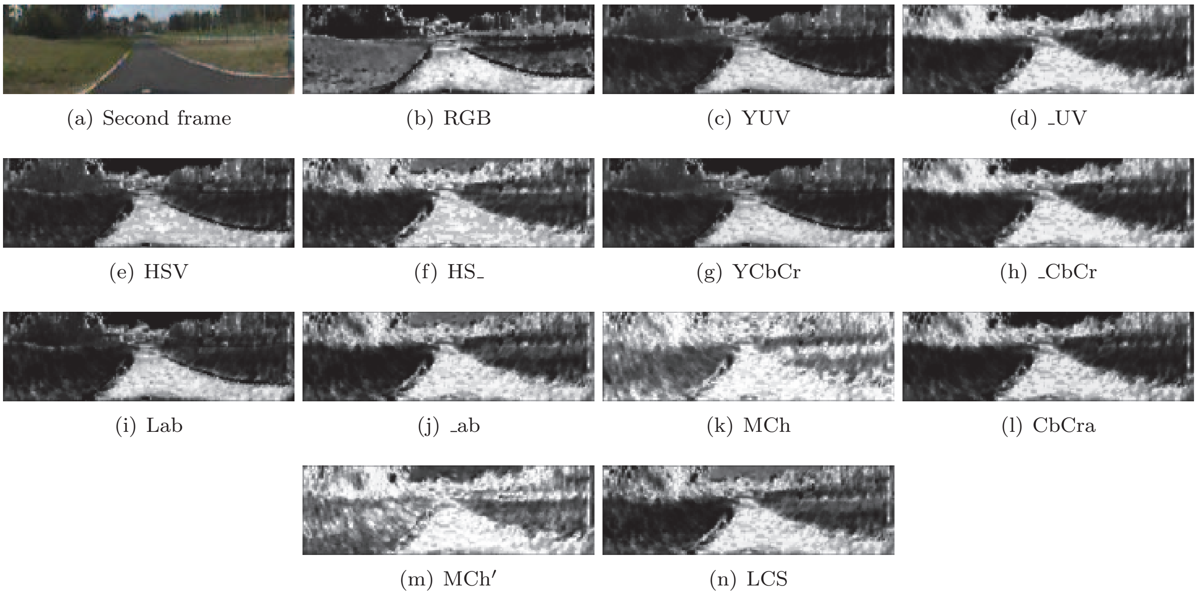


(a) Second frame (b) RGB (c) YUV (d) _UV (e) HSV (f) HS_ (g) YCbCr (h) _CbCr (i) Lab (j) _ab (k) MCh (l) CbCra (m) MCh′ (n) LCS

Figure 13: Shadows: Mahalanobis distance of the pixels of the second frame to the initial model derived from the first frame

The MCh and MCh′ colour transforms seem to perform badly in most cases from the point of view of the Mahalanobis distance, although MCh′, which uses the normalised colour information, performs better (apart perhaps in the case of Shadows). This could be due to the added b^* component, but given that CbCra performs better yet, the likely reason for the increase in performance is the normalisation.

The likely reason for the better performance of CbCra compared to MCh and MCh′ (despite encoding similar information) is the separation of the components. Indeed a single standard deviation is used for

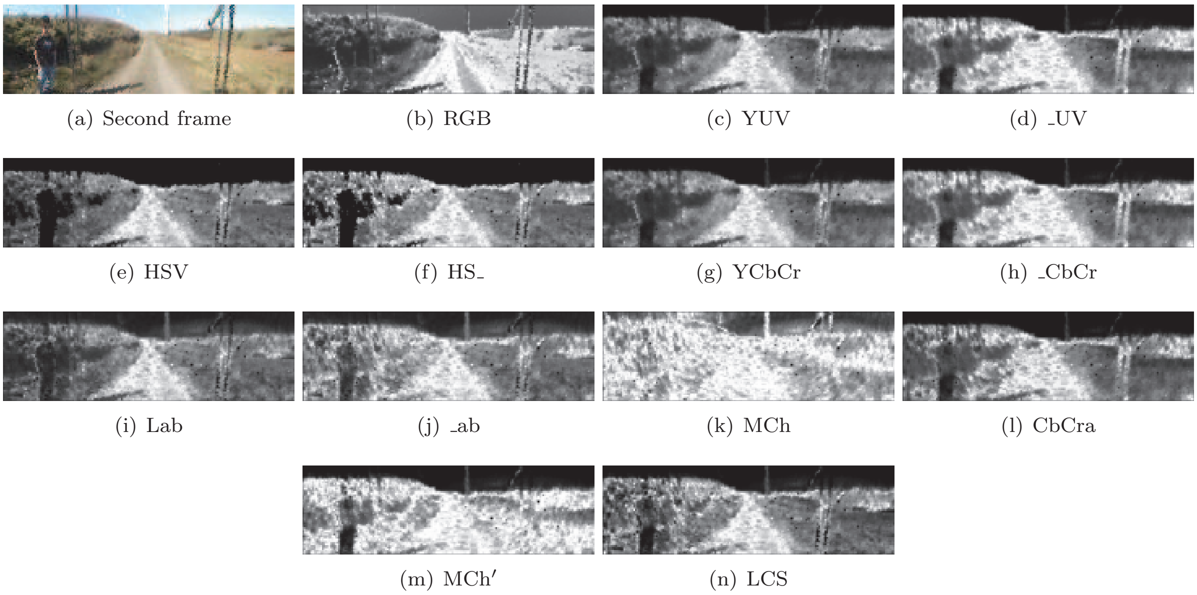

(a) Second frame (b) RGB (c) YUV (d) _UV

(e) HSV (f) HS_ (g) YCbCr (h) _CbCr

(i) Lab (j) _ab (k) MCh (l) CbCra

(m) MCh′ (n) LCS

Figure 14: FOOTPATH: Mahalanobis distance of the pixels of the second frame to the initial model derived from the first frame


(a) Second frame (b) RGB (c) YUV (d) _UV

(e) HSV (f) HS_ (g) YCbCr (h) _CbCr

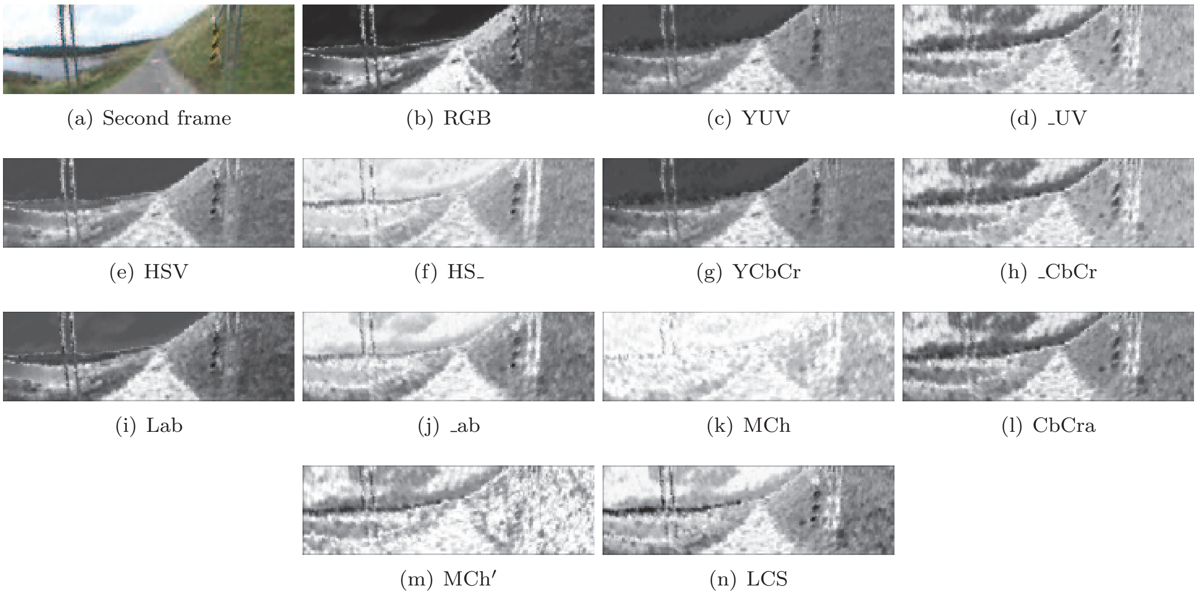
(i) Lab (j) _ab (k) MCh (l) CbCra

(m) MCh′ (n) LCS

Figure 15: WINDFARM: Mahalanobis distance of the pixels of the second frame to the initial model derived from the first frame

the combination of colour components and therefore the different variabilities of the various components is not taken into account. When combining the components, the normalisation helps, but only marginally compared to keeping the components separate.

The LLANBADARN dataset shows that in some cases, keeping the luminance information helps. Indeed, the colours in the images of this dataset are under-saturated and the main differences between road and non-road is the brightness of the pixels. However, and despite the noise pattern present in these images (due to


(a) Second frame

(b) RGB

(c) YUV

(d) _UV

(e) HSV

(f) HS_

(g) YCbCr

(h) _CbCr

(i) Lab

(j) _ab

(k) MCh

(l) CbCra

(m) MCh′

(n) LCS

Figure 16: Lakeside: Mahalanobis distance of the pixels of the second frame to the initial model derived from the first frame

a faulty camera), there is enough colour information for the road to be detectable, but only if the colour components are not combined.

After a qualitative evaluation of the various colour transforms, we now turn to a quantitative evaluation performed by running the proposed road following system on all the datasets with the various colour spaces. The analysis of the detection results provide more insight as to the performance of the system. In particular, they show that the system performs well for most colour spaces. However, there are failures and these are discussed for each dataset in turn.

3.2.2 Road detection

Tables 2 and 3 recapitulate the statistics on the position and width errors for all datasets and colour spaces. Of the two parameters, position is the most important one for autonomous driving given that the aim is to stay in the middle of the road. The tables show that for all datasets most colour spaces succeed with detection errors of only a few pixels. They also show that some colour spaces have errors that are significant, either in mean value of standard deviation. These global results are however only part of the evaluation. Indeed some of the colour spaces seem to fail (or not perform as well as others) from a statistical point of view but are not necessarily failures from a data point of view. Because the road detection was not restricted to the front part of the images, it does happen that it moves to detecting the road behind the robot, which makes the statistics show bad result while in fact the road is correctly tracked. Still, this shows that something went wrong at some stage, and we discuss now these cases. Further details on each of the datasets are given in Section A.1.

Figure 17 shows the position and width errors for three of the considered colour spaces for the Running-Track dataset. Typical failure cases on this dataset are due to yellow lines that cross the running track (_CbCr, HS_, MCh'), shadows and automatic brightness compensation of the camera (MCh', RGB). These result in the system detecting the road as including both running tracks (the correct light blue and incorrect dark blue) or only the wrong track. At frame 1506 a track branches onto the main running track and both RGB and Lab trackers include it in the main road. Lab eventually recovers but RGB does not (Figure 17)

Table 2: Mean and standard deviation (in pixel) of the position error for all datasets and colour spaces. Statistics for the Footpath dataset are calculated up to frame 1415 to avoid the junction at the end of the path.

	RunningTrack		Llanbadarn		Shadows		Footpath		WindFarm		Lakeside	
RGB	-15.1	60.9	-5.3	10.6	-92.2	63.4	-9.7	20.1	-30.8	29.8	-1.5	10.0
YUV	-2.0	2.0	-2.2	8.3	-12.2	2.1	-3.6	5.8	-48.3	35.9	-0.4	1.3
_UV	-2.9	3.3	-5.3	5.5	-12.1	2.0	-3.2	2.8	-1.8	9.5	-1.5	1.2
HSV	**-1.7**	**1.7**	-2.4	7.3	-11.8	2.1	-3.9	6.1	-50.3	35.2	**-0.2**	**1.4**
HS_	-3.0	4.5	-3.3	4.6	-11.2	1.8	-6.3	4.6	-48.1	41.0	-1.3	1.1
YCbCr	-1.9	2.0	**-2.2**	**8.3**	-11.9	2.0	-3.6	5.8	-48.1	35.6	-0.4	1.3
_CbCr	-2.9	3.3	-5.5	5.4	-12.3	2.1	-3.2	2.7	-1.8	9.5	-1.5	1.2
Lab	1.7	15.7	-3.0	7.6	-9.1	5.6	-7.3	13.8	-4.5	5.4	-0.4	1.6
_ab	-2.5	2.2	-3.9	5.2	-9.3	3.6	-4.4	3.6	**-0.1**	**3.6**	-1.6	1.1
MCh	-1.3	1.3	-7.6	7.8	**7.6**	**6.4**	-3.7	2.7	9.0	7.8	-45.3	159.3
CbCra	-2.2	1.1	-4.2	4.9	-11.9	2.0	**-3.1**	**2.9**	0.8	4.4	-1.6	1.2
MCh′	-6.7	10.0	-3.9	6.8	-35.7	21.2	-136.3	66.3	-38.1	35.2	5.3	10.1
LCS	-1.9	1.3	-2.8	4.5	-12.1	1.8	-3.8	2.5	-1.3	3.5	-1.3	1.1

Table 3: Mean and standard deviation (in pixel) of the width error for all datasets and colour spaces. Statistics for the Footpath dataset are calculated up to frame 1415 to avoid the junction at the end of the path.

	RunningTrack		Llanbadarn		Shadows		Footpath		WindFarm		Lakeside	
RGB	-1.5	10.6	9.7	14.2	**5.8**	**17.4**	18.5	14.1	-14.5	13.4	-4.5	10.2
YUV	2.3	3.7	5.4	13.2	14.2	8.5	12.7	11.7	-11.5	8.2	**-0.7**	**2.5**
_UV	-1.8	7.2	-16.4	10.2	13.3	8.2	8.9	8.3	-11.4	7.5	-3.5	2.8
HSV	1.8	3.6	**3.6**	**11.8**	12.2	7.9	12.3	11.8	-16.6	13.8	-1.3	2.7
HS_	-2.1	9.1	-10.9	8.4	8.7	7.5	**4.4**	**8.9**	-20.2	15.5	-3.8	2.8
YCbCr	2.2	3.8	5.4	13.2	14.0	8.4	12.7	11.7	-11.5	8.1	**-0.7**	**2.5**
_CbCr	-1.8	7.2	-16.3	10.2	13.8	7.8	8.8	8.4	-11.5	7.6	-3.5	2.8
Lab	1.3	5.1	4.6	12.9	13.9	9.1	13.9	13.1	**-5.4**	**5.3**	-0.9	2.5
_ab	-1.6	3.9	-10.9	9.4	12.4	8.1	7.1	9.1	-8.1	3.9	-3.1	2.7
MCh	-2.5	2.8	-33.4	11.6	-15.3	11.3	5.3	7.6	-25.6	13.1	-42.8	12.3
CbCra	0.3	2.7	-11.2	9.6	15.3	8.6	10.0	8.9	-7.9	5.1	-1.8	2.4
MCh′	-9.0	14.2	-27.5	10.8	-18.8	20.2	5.0	13.0	-30.1	14.5	-15.5	11.8
LCS	**-0.1**	**2.6**	-9.4	8.8	13.6	8.4	7.4	8.6	-6.6	3.7	-1.9	2.3

and the detection is pushed to the back of the robot.

Figure 18 shows the detection errors for three of the colour spaces for the Llanbadarn dataset. A systematic error to the right is visible, which is due to a muddy curb and yellow grass on the right. Despite this, the low quality of the images and obstacles such as raised pedestrian crossing, junction and sharp turn, all colour spaces succeed.

The detection errors for the Shadows dataset are shown in Figure 19 for three of the colour spaces. High contrast and image saturation means that the RGB colour space quickly includes the grassy area on the right as part of the road, leading to the failure of that colour space. A systematic error is also seen in this case, but this time is due to the ground truth including the shaded area of the road on the left while the initial detection was not because of the initial alignment of the robot. However, the MCh colour space does include this shaded area but not the bush area on the left despite being similar in terms of the Mahalanobis distance (see Figure 13(k)). However, the road is correctly detected because the expansion of the trapezoidal shape stops at the few pixels that do not match the colour model. This shows the strength of a shape constrained

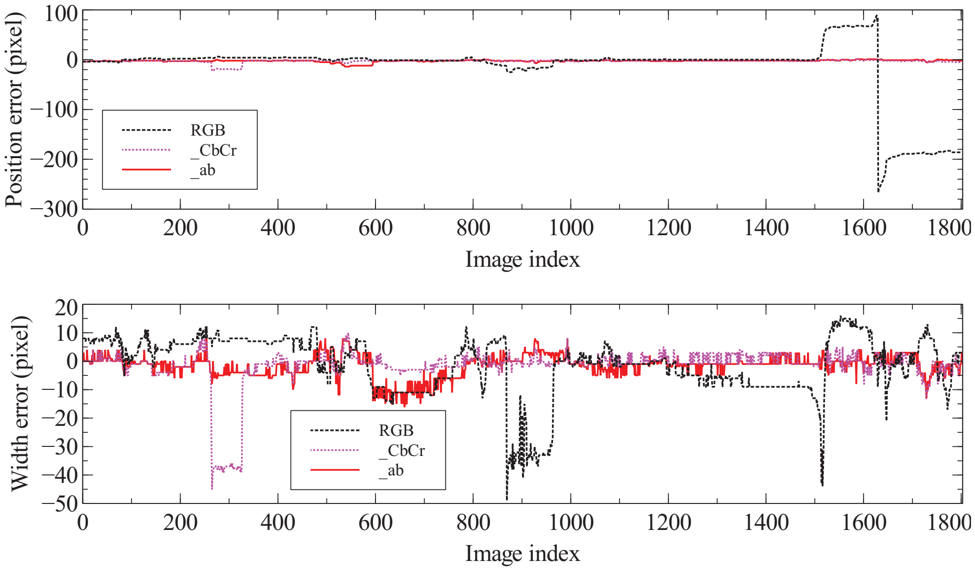


Figure 17: RunningTrack: position and width errors

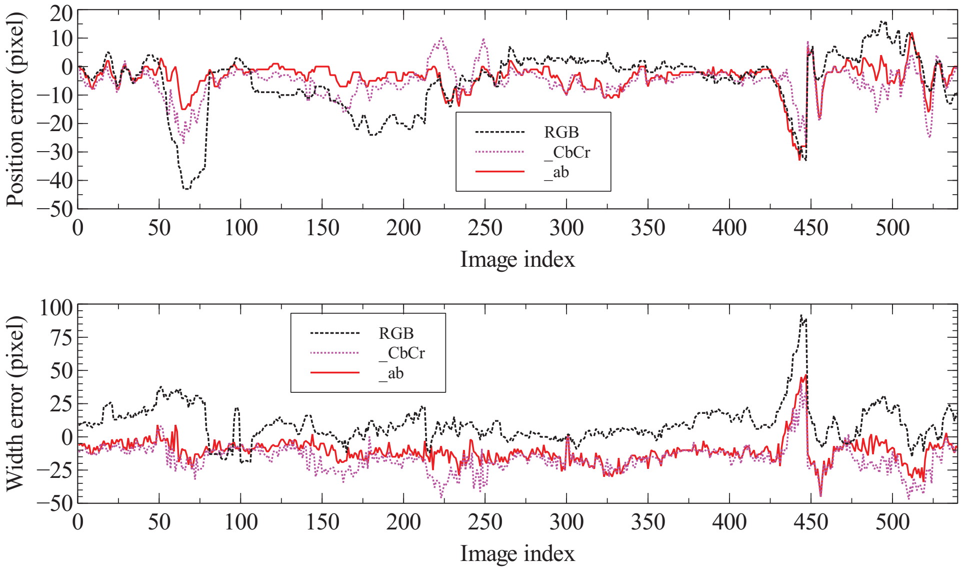


Figure 18: Llanbadarn: position and width errors

detection that is not matched to an explicit segmentation but expanded from previous knowledge of where the road is to be expected.

Figure 20 shows, for three of the colour spaces, the detection errors for the Footpath dataset, The road in this dataset presented a number of challenges, including a green cast on the left at the beginning and a number of wet patches. Combined with sudden changes in the width of the road and obstacles, this makes some of the colour spaces detect the road at the back of the robot approximately from frame number 1400. Moreover, the road branches off to the left at frame 1415 and the _ab detector, not wrongly, follows that branch. The statistics in Tables 2 and 3 do not include these final frames to avoid biasing the results because

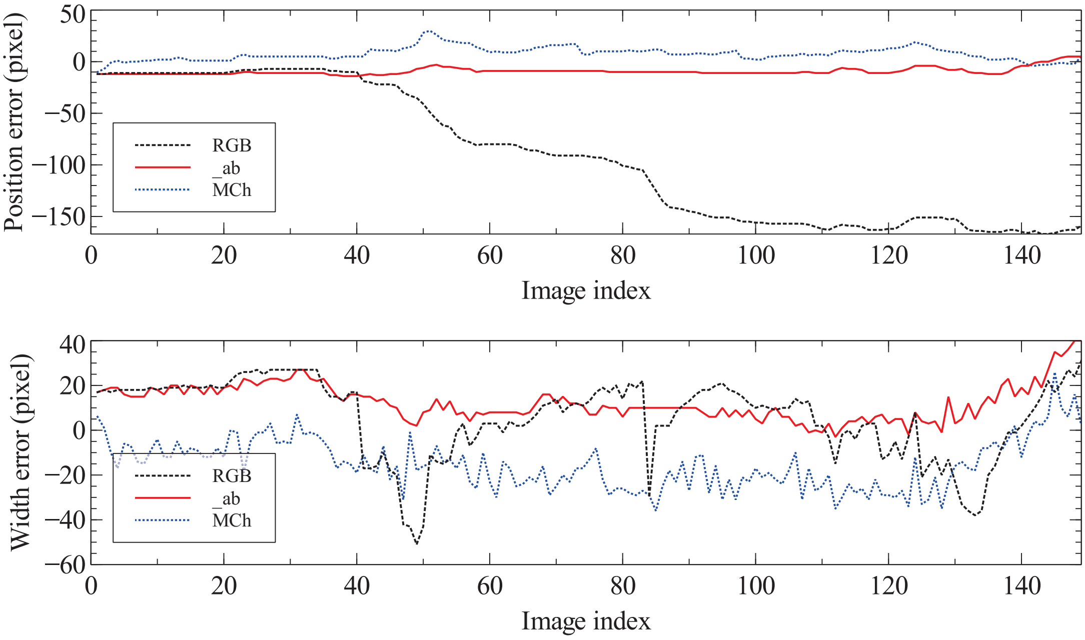


Figure 19: Shadows: position and width errors

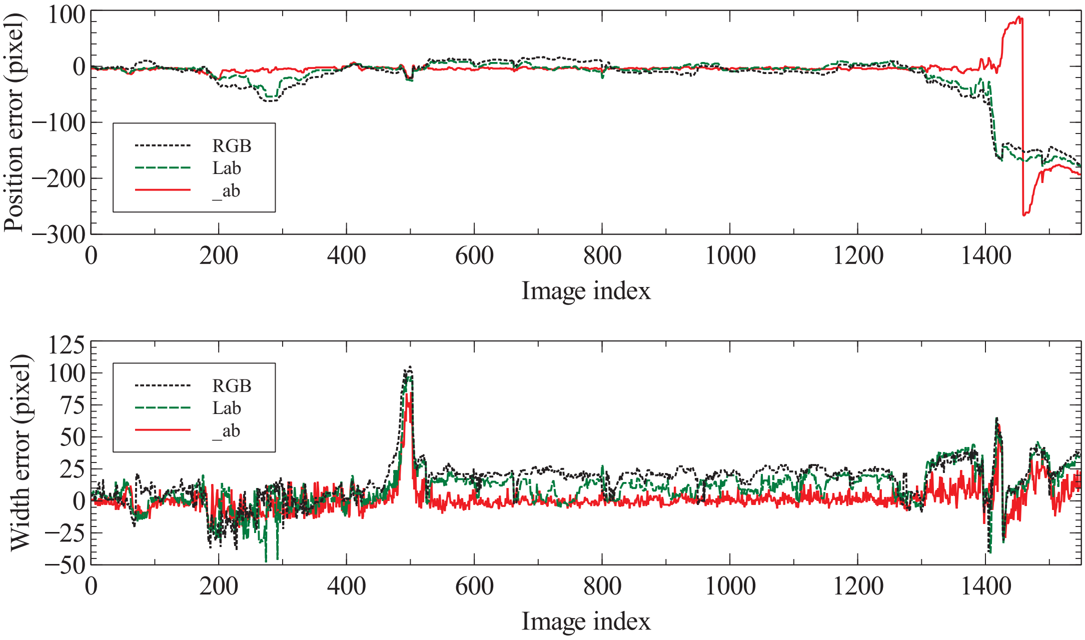


Figure 20: Footpath: position and width errors

the automatic detection followed a different branch than the ground truth.

The detection errors for two of the colour spaces on the WindFarm dataset are shown in Figure 21. This dataset presents many of the difficulties such a system can encounter: the road colour is not homogeneous (the centre being darker), changes as the robot moves both progressively as the nature of the ground changes

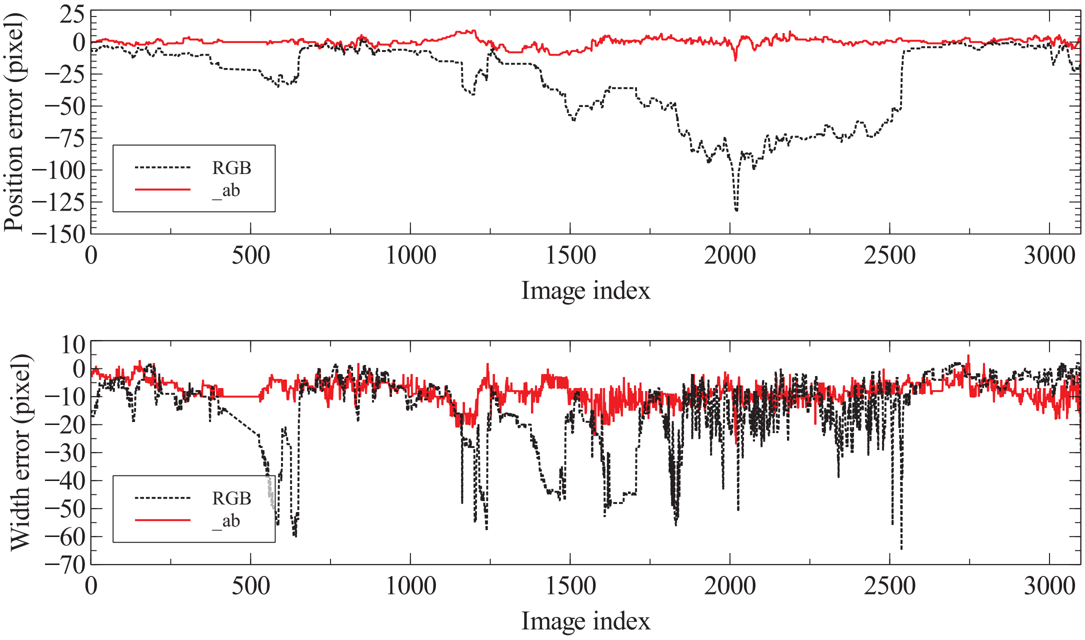


Figure 21: WindFarm: position and width errors

but also suddenly with shadows and wet patches. Moreover, the colour of the non-road areas are rather close to that of the road areas. Finally, because of the bad position of the robot on the road during image acquisition, the trapezoidal shape badly fits the data. However, the system generally performs well, with only the colour spaces that include luminance information failing at some point.

Figure 22 shows the detection errors for two of the colour spaces for the Lakeside dataset. In this case, all colour spaces succeed, despite the variable nature of the road surface and local changes due to wet patches. Only MCh fails from the beginning, which was to be expected given Figure 16(k). Contrary to the case of the WindFarm dataset, discarding luminance information has a negative effect (but only by 1 pixel for the position and 2 pixels for the width, no failures happening). This is because of the low colour contrast of the images due to a sometimes overcast sky.

3.2.3 Conclusion on colour spaces

From the discussion above, it is clear that the colour spaces that include luminance information are not as reliable over long runs, although occasionally they can help in separating well marked roads (which is usually not what happens in the field), especially in overcast days when images lack contrast.

The colour spaces that combine all their information into a single component do not perform well, especially when the road presents a variable appearance. This is because a single Gaussian is then used for the totality of the colour information, not allowing different variabilities. Normalising the various components before combining them helps, but does not cater for large variations.

The colour spaces that separate luminance information from colour information definitely perform better and the ones that do not retain the luminance information generally perform better yet and can cope with changes in the appearance of the road surfaces due to shadows or water. This property is partly due to the adaptability of the road model, but not exclusively, given that it is not enough for some of the colour spaces. Of all these colour spaces, _ab never fails and performs as well as if not better than other colour spaces, with

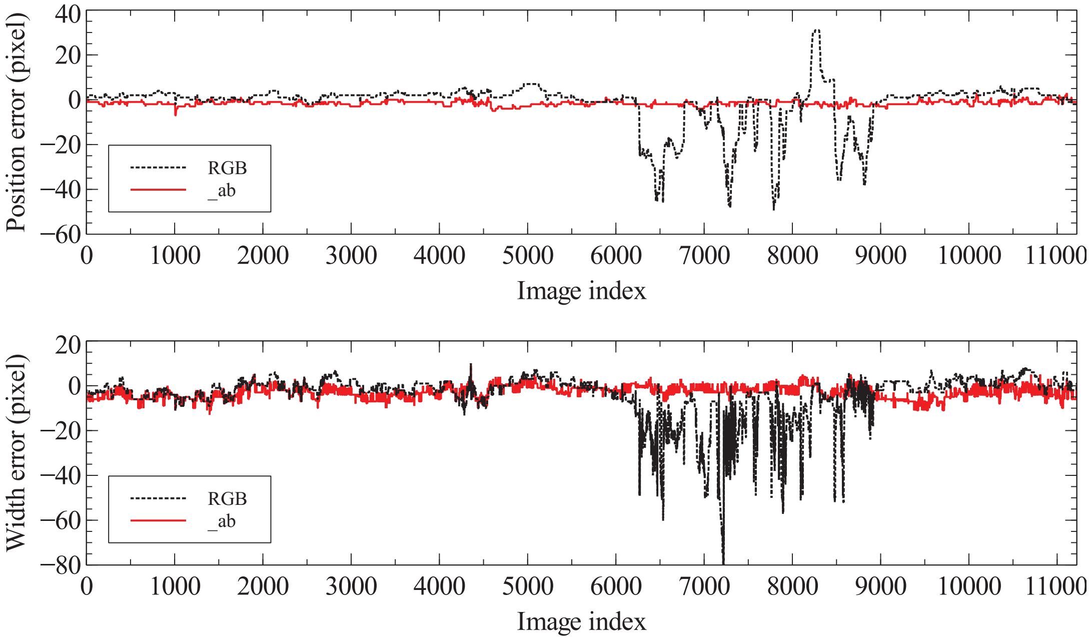


Figure 22: LAKESIDE: position and width errors

often a more consistent (lower standard deviation) detection (position and width). For this reason the _ab colour space is used for all subsequent results. Note however that no one colour space is better than the other and perhaps a better solution would be to offer a mechanism that would automatically select the best colour space based on the situation.

It is interesting to note that the *CIE L*a*b** colour space was designed to be perceptually linear, which is of little interest for this work. However, the colour information of this space performs (marginally) better than the non perceptually linear colour spaces.

Finally, if computational cost is an issue, then it might be better to use one of the colour spaces that is a linear transformation from RGB, and _UV or _CbCr would be good candidates. Moreover, some cameras directly provide images in colour spaces such as YUV, so _UV might be a good choice in such cases.

3.3 Adaptability evaluation

We now show the effect of the adaptability factor (φ in (4)). For this we ran a number of the datasets presented in Section 3.1 through our system, varying φ from 0 (no adaptability) to 0.5 (the experiments reported in Section 3.2 used $\varphi = 0.05$). For these values, we also sub-sample the datasets, taking every 1 (all), 2, 3, 4, 5 and 10 images to simulate the effect of varying processing and/or robot speed. The results are reported in Section 3.3.1.

The simulated varying robot speed obviously does not take into account the dynamics of the system. To test that we have repeated the same straight section at the beginning of the road where the LAKESIDE dataset was captured (from Point 1 towards Point 2 in Figure 26) at varying speeds. The results are reported in Section 3.3.2.

3.3.1 Varying the adaptability factor

We show here some of the datasets used previously, in a different order to help the presentation.

Figure 23 shows the mean errors on position and width for different values of the adaptability factor and speed for the Lakeside dataset and the _ab colour space. It clearly shows that increasing the adaptability factor, for a given speed, improves the detection of the road. More specifically, the width detection improves, while the position remains sensibly the same as long as any adaptation occurs. This shows that adaptation allows better detection of the road, the width being too large with not enough adaptation and converging towards an error of about 2 pixels for lower speeds. The position error is largely independent of the adaptability; this is because the surroundings of the road are symmetrical, both sides being grass. If that had not been the case, the position error would have also depended on the adaptability factor. These results also show that space sampling (speed of the robot versus processing speed) somehow affects the performance. For slower sampling, the system needs higher adaptability to reach similar performance. This is because changes in the colour properties of the road take fewer frames to enter the trapezoid at faster speeds, therefore requiring higher values of the adaptability factor to keep track of the new colour values.

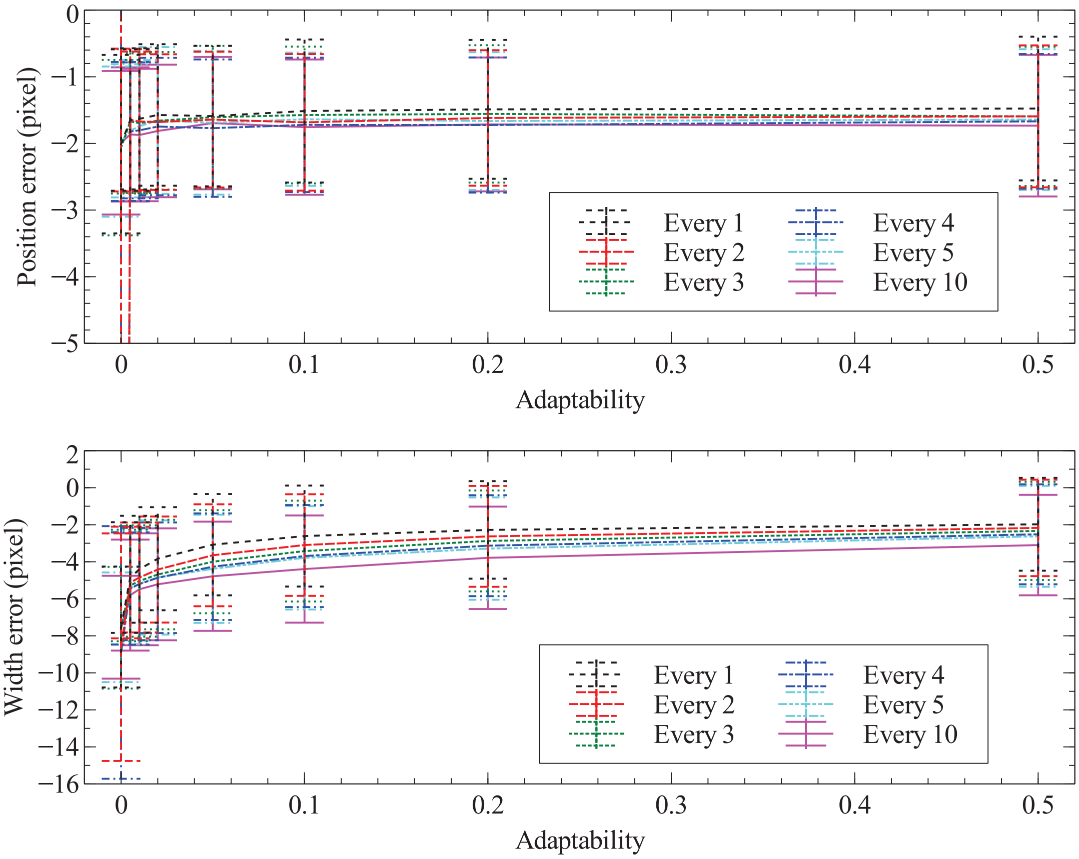


Figure 23: Lakeside: mean position and width errors for different adaptability factors and speeds using the _ab colour space

Figure 24 shows the statistics for position and width of the detection on the RunningTrack dataset with the _ab colour space. Compared to Lakeside, the results on this dataset are not so obvious. Again the position is largely independent from the adaptability factor and speed. However with a too low adaptability factor the width tends to be overestimated but not so much with higher adaptability values. This is because the changes

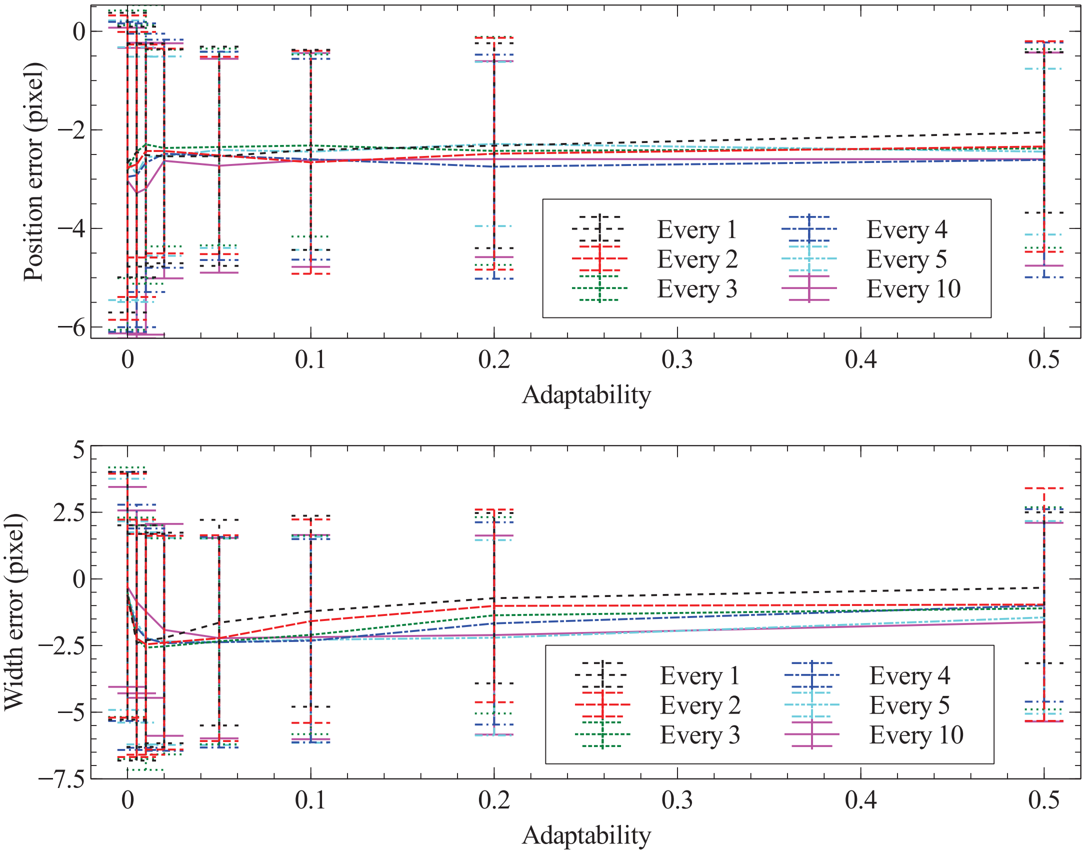


Figure 24: RunningTrack: mean position and width errors for different adaptability factors and speeds using the _ab colour space

in the images correspond to sharp shadows and sudden brightness adjustments of the camera, changes that are too fast for low adaptability values. The performance is better without adaptability compared to too low adaptability because the initial model remains correct given the colour space used (_ab) and the fact that the road is well delimited by a white line. This is not the case using the RGB colour space (Figure 25) where no adaptability leads to a much worse performance. The poor performance with no adaptability is due to the detection jumping on the darker lane at the various transitions into and out of the shadow areas.

Close examination of the detection results shows that for Every 1, up to an adaptability factor of 0.01 the results are good for the RGB colour space (and performs better in shadows compared to not using any adaptability). Moreover, the low adaptability value means that the bank on the side of the track at the end of the dataset is not learnt and the detection remains on the track. Higher adaptability values perform similarly in the shadows but learn the bank as being the new road. Finally, the detection at higher speeds of the robot performs similarly but for higher values of the adaptability, showing the relationship between adaptability and frame rate.

3.3.2 Varying the driving speed

We tested the system on the Idris robot at different actual driving speeds on the road used for the Lakeside dataset, from Point 1 towards Point 2 in Figure 26. This experiment was performed with the default adaptability value of $\varphi = 0.05$ and the robot was driven at 1 m/s while logging data (including images). The

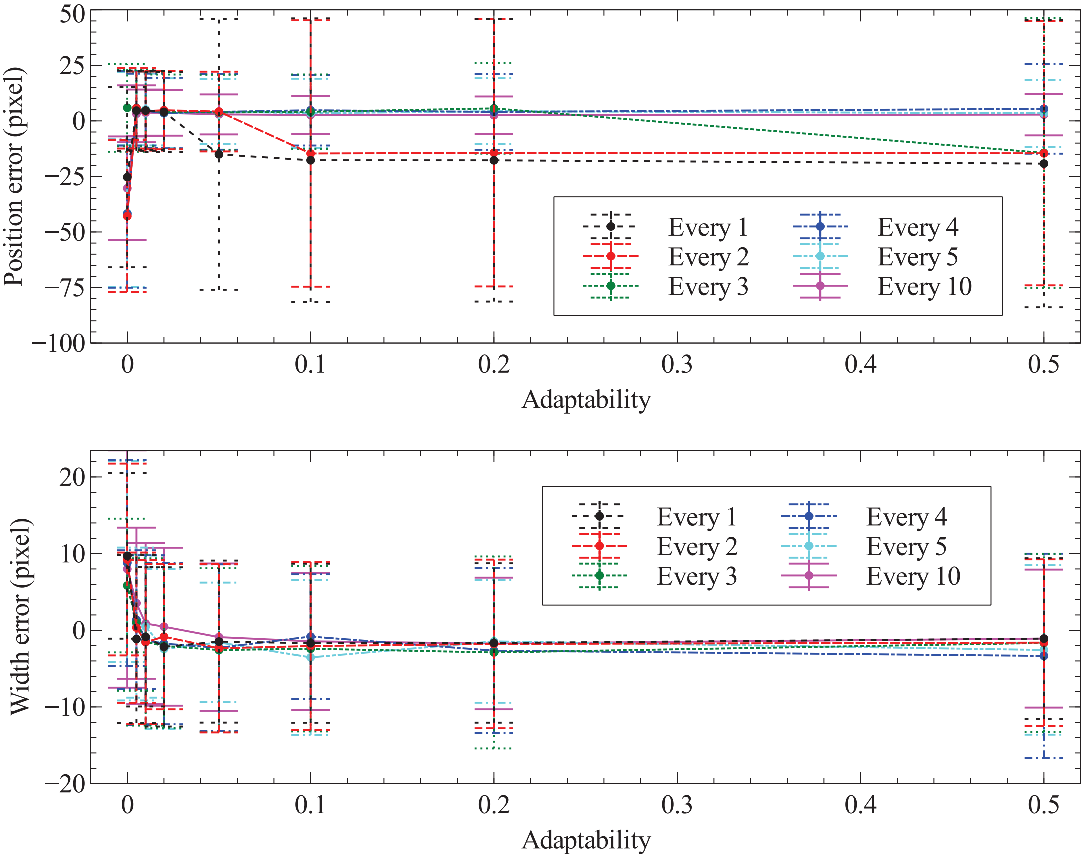


Figure 25: RunningTrack: mean position and width errors for different adaptability factors and speeds using the RGB colour space

results for that speed were presented in Section 3.2.2.

We repeated the same road section at various speeds from 0.5 m/s. At a speed of 2 m/s, large oscillations in the driving started appearing, the amplitude increasing. Note that this was not due to the road being lost but due to the simplistic and badly tuned proportional controller used to drive the robot, coupled with the frame rate. Not doing any data logging (which roughly doubles the frame rate), driving at the speed of 2 m/s was successful but the system started failing again at 3 m/s.

Another effect of varying the speed was experienced during two runs. At Point 4 in Figure 26 the road is interrupted by a gate, on the path of the road, and a cattle grid on its left. The road therefore presents a sudden change in appearance (colour and texture) directly in front of the robot and high up (above the ground) where the gate is and on the ground where the cattle grid is. Having started at Point 3 driving at 1 m/s the system finally failed when reaching the cattle grid. The robot correctly turned away from the gate towards the cattle grid but failed to adapt to the new colours given by the cattle grid and therefore failed. However, on a repeat run (from Point 1) when slowing down on the approach to about 0.5 m/s, therefore allowing more adaptability per distance travelled, the system succeeded in passing the cattle grid. Images corresponding to these events are shown in Section A.2.

These experiments show that the adaptability factor depends not only on the frame rate but also on the speed of the robot. The performance is therefore dependent on the ground covered between images. The experiments reported tend to show that with an adaptability factor of 0.05, processing at 3 frames per metre

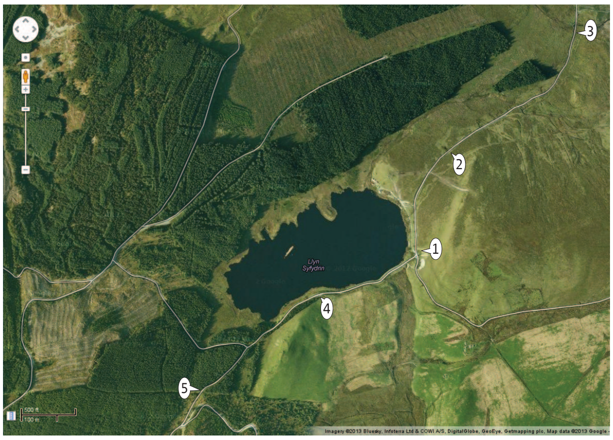

3
2
1
Llyn Syfydrin
4
5
500 ft
100 m
Imagery ©2013 Bluesky, Infoterra Ltd & COWI A/S, DigitalGlobe, GeoEye, Getmapping plc, Map data ©2013 Google -

Figure 26: Location of the longer field trials

Table 4: Length of the various long-range trials (points are from Figure 26)

Points	Distance (m)	End?
1 – 3	850	Stream
3 – 4	1156	Cattle grid failure
4 – 1	306	Return to base
1 – 5	740	Puddle failure

travelled is a good frame rate. Different values of the adaptability factor would lead to different frame rates.

3.4 Repeatability and long-range driving

Most of the roads that were used to create the datasets used in the paper have been driven many times by the system. Apart from the failure cases describe elsewhere in the paper, the system deviated very little from the centre of the road. Specifically, we drove the section of road corresponding to the Lakeside dataset a number of times, North to South and South to North, at speeds of 1 m/s and 1.5 m/s. The robot never deviated from the road which was approximately 1.5 times wider than the robot, with a repeatability much better than that of standard GPS receivers. This section and other sections of our long-range field trials were repeated several times.

Longer distances were also driven by the system. In most of the long-distance trials we did (similarly to the shorter datasets presented in Section 3.1), what stopped the system was the presence of an obstacle that could not be passed by the robot, apart from two cases described below. Some of the recorded tracks are described in Table 4 and Figure 26. The cattle grid failure was described in Section 3.3. The puddle failure was due to a combination of image saturation and coincidence of colours on- and off-road in a bend of the road, a situation that would have been difficult for a human being (see Section A.3).

4 Discussion and conclusion

Following *roads* (as defined in Section 1) should be performed reliably with all variants of roads ranging from streets to footpaths. We have shown that the method we proposed in this paper is capable of following a multitude of roads in a robust and reliable way. The method requires a single frame to learn the road appearance and makes a simple assumption about the roads: that their colour can be distinguished from that of the surroundings of the road. Most well used roads are of that type, given the appropriate colour space. Adaptability ensures that the algorithm is robust to environmental factors and road colour changes. Coupled with the use of the _ab colour space the method copes well with local road appearance changes such as that due to shadows and wet patches.

Methods that first segment the images into road/non-road regions and then fit a model of the road on the result can fail if the boundary between the two areas is ill-defined (for example the case of the Llanbadarn dataset), especially since often the segmentation uses a fixed threshold on some colour similarity measure, as in (Thrun et al., 2006). Indeed this could create large areas of the image classed as road when they are not, leading to a bad fit of the model. Our algorithm that increases the width of the road stops at the edge of the road as long as some part of the edge of the road are defined along the extent of the shape. Moreover, the expansion of the model is not stopped at any of the small variations on the road because whole lines of pixels along the road are considered at a time, therefore low-pass filtering the pixel noise. Significant changes in the value of the distance are only (mostly) introduced when the edge is reached.

Adaptability is done progressively (to avoid learning off road colours in case of bad detection) in two ways. One is that because of the tapered shape of the model, new road colours only appear progressively in the view and therefore only slowly influence the colour statistics which are dominated by that from what is assumed to be free space (the area immediately in front of the robot). Second, to explicitly control the adaptability, we introduced a damping in the learning that proves effective and offers a parameter that can be changed depending on the speed of the robot (and values based on our experiments were given).

The combination of the distance and road width terms in (2) proved to be enough, despite its simplicity. More complex combinations could have been considered, but we generally aimed at simplicity for efficiency reasons because this system is only intended to be a part of a general navigation system and therefore needs to be fast. In particular, the model of the road was also chosen for its simplicity because it allows the easy and efficient calculation of pixel positions. Indeed, the model is made of a base set of pixels that correspond to the minimum width model and can be enlarged by adding one (diagonal) line of pixels at a time. All the coordinates can therefore be pre-computed and shifted to address different lines.

The various parameters of the model depend only on the geometry of the system, or are established as part of the road detection. We have manually set the former based on the robots we used in this work. However they could have been established automatically by placing the robot in the correct position on the road and fitting the model to the road in a similar way as we presented but taking into account all the parameters. However, the value of these parameters is not critical. Finally some of the parameters, such as height of the trapezoid, could be adjusted as part of a wider navigation system to take into account desired speed for example.

As the model is horizontally expanded to find the edges of the road, the statistics of the colour are gathered to compare them to the road model. The results of the calculation is cached so that no further evaluations are needed to update the colour of the road once it has been detected, even when only using a narrow version of the shape for statistics update. This saves precious time and allows the method to run on-line at speeds the robot can handle, even on a 800 MHz Pentium® III computer with 256MB of ram. This caching is possible because the distance function (3) is fully evaluated only once, which is performed incrementally. The geometrical expansion of the model assumes that the starting point is contained within the road area of the images. This is not an unrealistic assumption if the frame rate is fast enough given that the initial model is narrower than the roads. However, should this not be the case, a viable solution would be to start the detection procedure by first shifting the model to see if a better match is obtained. This would move the initial model in the valley of the distance function, before expanding it.

The shape used to model the geometry of the road does not necessarily fit the road well, in particular when the road is not flat, straight or of constant width. When the road presents a turn, the tip of the trapezoid pushes the detection sideways thanks to the horizontal expansion of the model during its construction, especially if the road is wider than the minimum width (when the base of the trapezoid is too narrow).

This allows the robot to start the turn as soon as the road bends. The height of the trapezoid is therefore directly linked to the necessary forewarning needed for the robot to be able to follow the turn. This same construction method of the model ensures that the model is fully enclosed in the road area, therefore only representing the effective straight part of the road to follow. This is an advantage over fitting a model in some best fit sense that would tend to include in equal amounts road and non-road areas around their boundary, therefore wrongly reporting a wider road.

Our proposed method relies on the assumption that the road is distinct from its surroundings based on colour only. This only excludes roads that are solely defined because multiple vehicles have travelled in the same tracks in an otherwise open area (such as in the desert). Often such tracks result in the “road” being of slightly different colour than the non-traversed areas and it is possible that this difference is enough for our method. We have however not been able to evaluate this.

The reliance on colour differences and the simplicity of our method makes it an ideal candidate to objectively compare multiple colour spaces. We have therefore presented results on multiple datasets using a number of colour spaces, many often used in the literature (with or without justification). Our results show that the colour spaces that discard luminance information often perform best because they offer resilience against changes due to shadows and local surface properties such as wet patches. Although the literature often shows the use of colour spaces that separate luminance from colour information, the discarding of luminance information is seldom used. Moreover, combining multiple colour components (possibly from different colour spaces) into a single component as proved to produce worse results.

Many authors have used mixtures of Gaussians to represent road colours (e.g. (Thrun et al., 2006)). However, results given in (Lee and Crane III, 2006) on a very small (and therefore possibly not representative) dataset show that using two Gaussians instead of one for the road colours does not improve the results dramatically. The proposed method only uses one Gaussian on several grounds. The first is that handling a single Gaussian is easier and faster. The second is that some of the colour variation is efficiently handled by colour transformations, as the results in Section 3.2 show. The third is that multiple Gaussians are only needed when the road is genuinely multi-coloured, which on ill-defined roads happens when the centre of the road (between wheel tracks) is of a different colour. However, this colour is often the same as that surrounding the road and including such colour in the model would therefore be detrimental, or at least wasteful if the pixel classification method detects that some of the on- and off-road colour properties are the same and does not use them. Finally, the images we used were of low resolution and the shapes typically contained approximatively 750 pixels (for a width w of 20). Calculating more than one Gaussian distribution on this number of pixels is becoming close to not being significant.

Perhaps a better solution to the problem of the centre of the road having a different colour would be to ignore this area, or weigh it down in the computation of the colour statistics. From a computation point of view this would be easy and therefore not time consuming. However, we have not done this as the reduction of the number of pixels was too great to keep the statistics significant. The use of a higher resolution camera would easily allow this.

Robust statistics (such as median and median absolute deviation) could have been used to represent the road colour. However this is generally not done in the domain of road following for efficiency reasons. Moreover, the main problem such approach would solve is robustness against outliers, which is not the main problem encountered here. In particular, this would not improve the colour representation using a single Gaussian when a central (often green) area is present on the road as this is genuinely a distribution corresponding to a different mode rather than outliers. Moreover, our incremental fitting of the model allows us to compute the mean as the model is fitted, which would not be possible should a median be used.

It is not clear yet if other features of the road surface could be used. Some published work has used the gradient of the image as a measure of texture or more explicit texture properties such as the Haralick statistical features. However, it is not obvious that such features would offer more discrimination on some of the types of road we have used for this work. Perhaps more importantly would be to consider some rough 3-D information, perhaps captured by a laser scanner (such as the one used here for safety) as road surroundings often present tall grass or features that are different from the road surface. We will evaluate such aspects in future work.

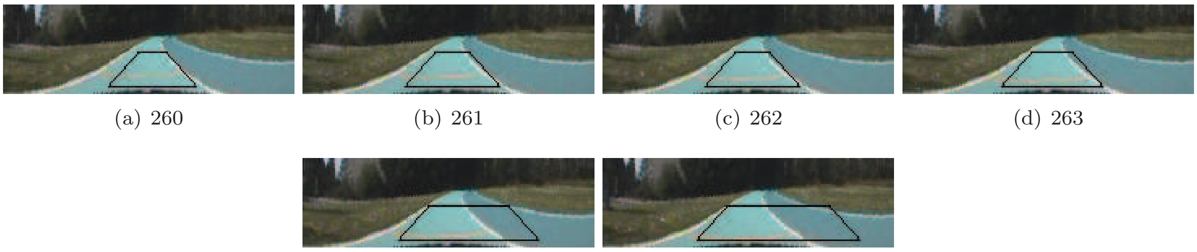
A Detailed description of results

We present in this appendix detailed descriptions of some of the results presented in the paper, with accompanying figures. These are intended as a way of showing the various successes and failure modes of the method.

A.1 Colour spaces

A.1.1 RunningTrack detection

The detection graphs of the RunningTrack dataset (Figure 17) show typical failures with different colour spaces. The first type happens at frame 260: a yellow line across the track in the otherwise homogeneous colour pushes the detection to span both running tracks. A subsequent yellow line in frame 325 in the other track pushes the detection back on the correct track. This happens for the _CbCr and HS_ colour spaces. Figure 27 show the transition frames with the detected road drawn on the images. The same failure happens at frame 429 for HS_, which then recovers promptly. The same also happens to MCh′ at frame 429 but this colour space does not recover so quickly. In fact shadows cast by nearby trees push the detection even further in the other lane. This is due to a combination of the shadows on the ground and an overall change in image brightness due to auto adjustment of the camera when the robot entered the shadow. Figure 28 shows the frames corresponding to that failure. The overall change in brightness of the images can be seen between frames 507 and 508. A similar effect happens in the RGB colour space, but when exiting the shadows, at frame 867. In the shaded area, other colour spaces have shown variations away from ground truth, but only by a few pixels.



(a) 260

(b) 261

(c) 262

(d) 263

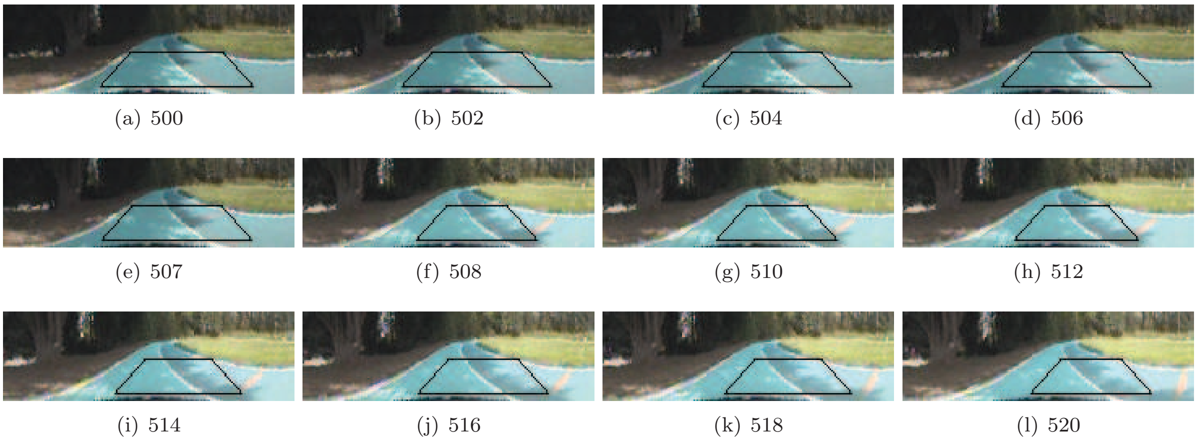
(e) 264

(f) 265

Figure 27: Failure case of the _CbCr colour space on the RunningTrack dataset: a yellow line (visible in the colour version of the paper) pushes the detection to span both running tracks

At frame 1506, a running track of the same colour as the one the robot tries to follow branches on the track from the left. Both RGB and Lab trackers expand to include this added available width, despite the white line separating them and then latch onto the joining track only (and the detected road is therefore too narrow, because of the perspective). At frame 1628, when the joining track finally fully merges with the main track, the Lab tracker moves back to the main track. However, the RGB tracker keeps tracking the side of the main track and eventually switches to tracking the road at the back of the robot. The RGB case is shown in Figure 29.

The road being tracked at the back of the robot happens with other datasets (see below). When this happens,

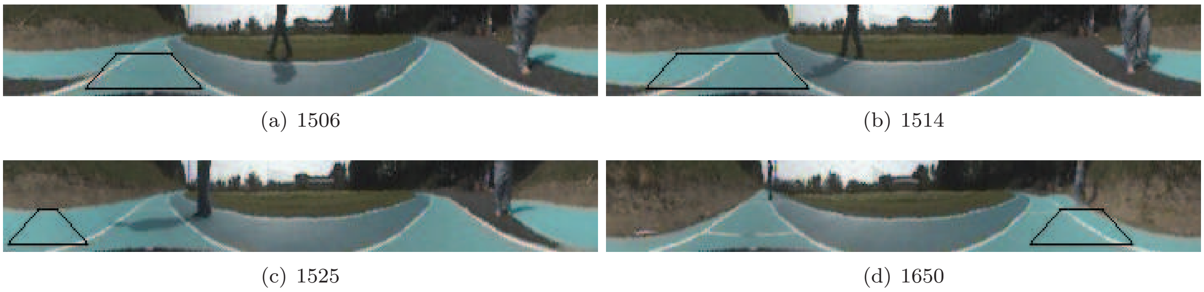


(a) 500 (b) 502 (c) 504 (d) 506

(e) 507 (f) 508 (g) 510 (h) 512

(i) 514 (j) 516 (k) 518 (l) 520

Figure 28: Failure case of the MCh′ colour space on the RunningTrack dataset: the detection is pushed away by shadows onto the other running track



(a) 1506 (b) 1514

(c) 1525 (d) 1650

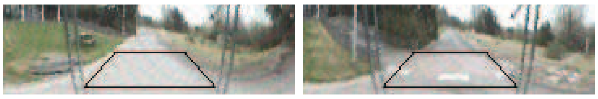
Figure 29: Failure case of the RGB colour space on the RunningTrack dataset: the detection moves onto a joining track on the left and then moves on the back of the robot

the statistics on the errors (mean and/or standard deviations) tend to be very large. The statistics in these cases, although correct, are not necessarily indicative of bad performance and in most cases the back road is tracked correctly. However, this is indicative of "something going wrong" at some point, often due to intersections.

As visible in Tables 2 and 3, for the RunningTrack dataset, apart from the RGB, Lab and MCh′ colour spaces, all the colour spaces perform adequately, despite the shadows, track discolourations and intersection.

A.1.2 Llanbadarn detection

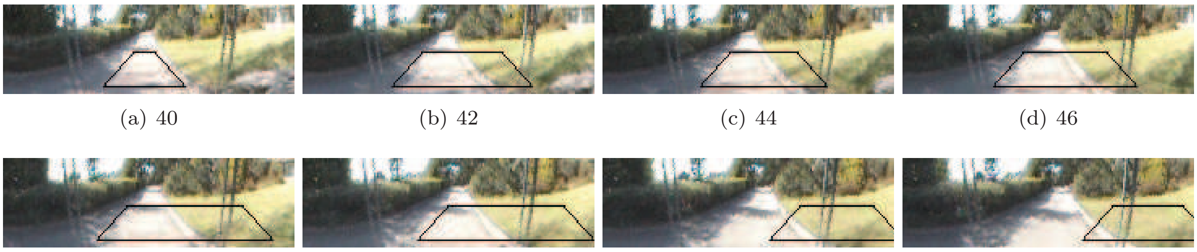
For the Llanbadarn dataset, it is not surprising that the detection results appear to be noisy (Figure 18). As the Mahalanobis images (Figure 12) show, the road is only weakly visible in most colour spaces. It is interesting to note that the road is systematically detected too far to the right. This is because the right side of the road is only weakly marked by a muddy curb and yellow grass while the left hand side is well marked by green grass, see Figure 30(a). Despite the quality of the data and the obstacles present (raised pedestrian crossing, Figure 30(b)), the errors are reasonable in many colour spaces.



(a) Frame 180: the road is not well marked on its right side and therefore systematically detected too far to the right

(b) Frame 215: the raised pedestrian crossing is correctly detected

Figure 30: Road detection in _ab colour space for the Llanbadarn dataset



(a) 40 (b) 42 (c) 44 (d) 46

(e) 48 (f) 50 (g) 52 (h) 54

Figure 31: Failure case of the RGB colour space on the Shadows dataset: the detection moves on the grass area next to the road and then keeps tracking the grass

A.1.3 Shadows detection

The Shadows dataset (Figure 19) shows two failures. Using the RGB colour space, the detection suddenly, at frame 41, includes part of the grass next to the road. This happens just after coming out of the shaded area where the images become very saturated, therefore not showing enough distinction between the road and the grass in the RGB colour space. Figure 31 shows some of these frames. Later, the detection is pushed further back by dark bushes, explaining why the road is then detected behind the robot. The same effect is seen using the MCh′ colour space, although not so dramatically initially.

Apart from these two cases, the detection appears to be always too narrow and too far to the right. This is because the ground truth was set to include the whole width of the road, including the shaded area on the left. The initial model did not include that shaded area and was therefore not detected as part of the road. This is a failure of the initialisation that only grows a symmetric model, which is a good procedure, assuming the robot is centred and aligned with the road which was not the case for this dataset.

The only colour space that includes the shaded area is MCh. Given the Mahalanobis image in Figure 13 this is not surprising. What is surprising is that the detection remains on the road given that the whole bush area on the left of the road and grass area on the right appear to be similar to the road. However, the matching is constrained by the trapezoidal shape and its expansion stopped as soon as making it wider increases the accumulated Mahalanobis distance. A few pixels not matching well are enough to stop the expansion process, offering robustness against such bleeding of the expansion.

Again, apart from the RGB and MCh′ cases all colour spaces perform well and similarly.

A.1.4 Footpath detection

The detection results for the Footpath dataset (Figure 20) show more of the previous failures. At frame 186 the detection is pushed towards the grass on the right of the road by shadows cast by nearby trees, this for

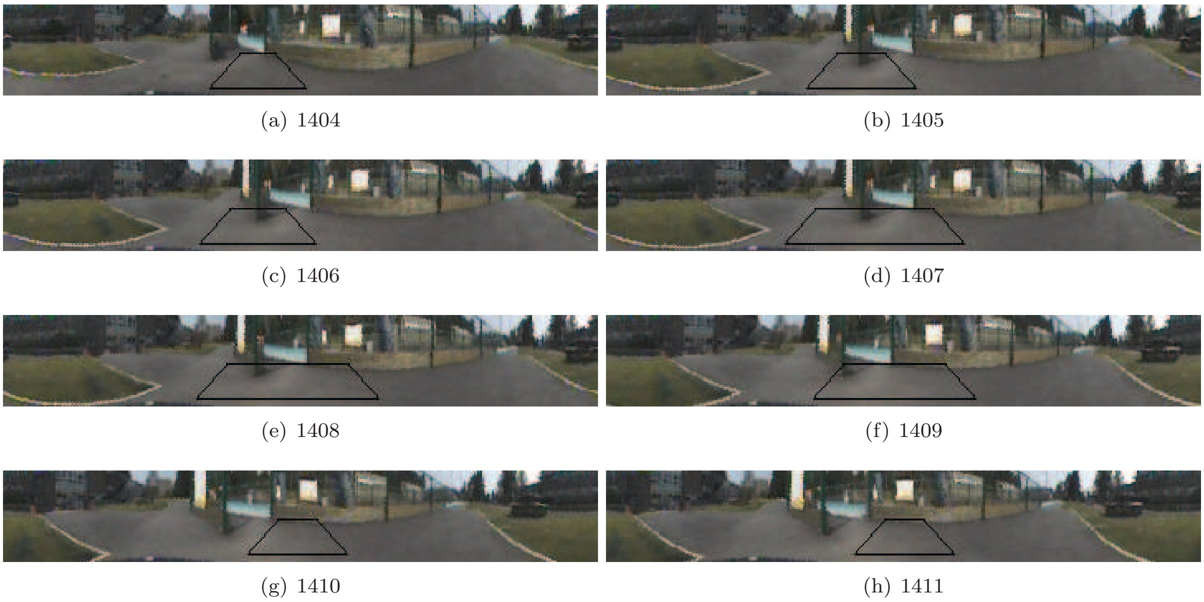

(a) 1404 (b) 1405 (c) 1406 (d) 1407 (e) 1408 (f) 1409 (g) 1410 (h) 1411

Figure 32: Failure case of the Lab colour space on the Footpath dataset: the detection fails to detect the whole width of the road and is pushed behind the robot by an sudden obstacle on the right

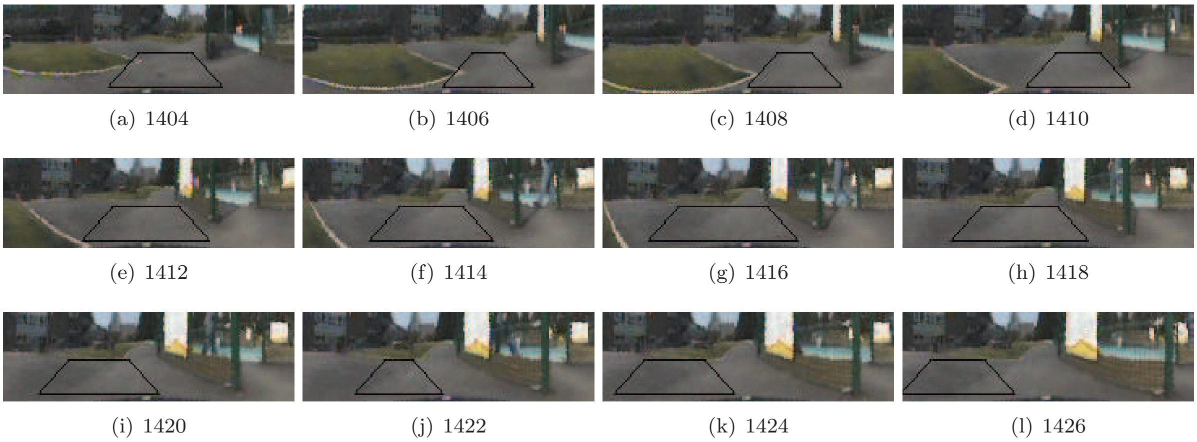

(a) 1404 (b) 1406 (c) 1408 (d) 1410 (e) 1412 (f) 1414 (g) 1416 (h) 1418 (i) 1420 (j) 1422 (k) 1424 (l) 1426

Figure 33: Failure case of the _ab colour space on the Footpath dataset: the detection follows a wider road branching off on the left

the RGB, Lab and MCh′ colour spaces. Only MCh′ does not recover from this and ends up detecting the road at the back of the robot.

At frame 1302, a widening of the road is not detected using the RGB and Lab colour spaces, due to slight variations in colour of the tarmac due to wetness. This leads the system to detecting the road too far to the right, against a fence. At frame 1405, an obstacle on the right (a door opened in the fence) pushes the detection further to the right and eventually behind the robot. This is shown in Figure 32.

Just after that obstacle, a wider road forks out to the left from the straight road. This is the one the detection using _ab follows, creating the large error seen for that colour space, Figure 33. Note that this is the only case where the _ab colour space has not followed the ground truth. Even so, this case is not really a failure as a plausible road, of the correct colour, was followed. Without this event, _ab has similar statistics as the other colour spaces, Tables 2 and 3.

A.1.5 WindFarm detection

The WindFarm dataset is the most difficult one in that the road is not well defined, being made of gravel and surrounded by tall yellow grass. The appearance of the road changes now and then due to wet patches and (reflective) water puddles. Also, in many parts the centre of the road is significantly darker than where the wheels drive, as is often the case on such surfaces.

This dataset also suffers from the fact that the robot was not at the centre of the road. This is because the dataset was captured during a trial of an early version of the system that had an offset in the expected road position. The robot being close to the side of the road, the safety laser scanners were often tripped by the presence of long grass overhanging in front of the robot, therefore stopping it, resulting in long sequences where the robot does not move, while the grass is being moved away. While pushing the grass away, the operators did cast shadows in front of the robot, which on occasion was enough to push the detection away from the front of the robot, especially using the colour spaces that contain luminance information. The effect of the robot not being centred on the road is that the trapezoidal shape does not fit the data as well as it should, for large parts of the dataset.

All these difficulties combined to make the detection results noisy (Figure 21). Most colour spaces that include luminance information eventually fail, detecting the road in the grass on the right of the road, and sometimes recover. In terms of road position, _UV, _CbCr, _ab, CbCra and LCS all perform very well, _ab having the lower mean error and standard deviation. In terms of width, Lab performs best but _ab is more consistent (lower standard deviation).

A.1.6 Lakeside detection

The results (Figure 22) on this dataset are not surprising. Most colour spaces perform well. However, MCh performs very badly from the first frame (with a too large width) and does not recover, the positions and width then becoming random (the statistics for this colour are therefore meaningless). This is not surprising given the Mahalanobis image in Figure 16. In both the MCh' and the RGB cases, the errors come from the model becoming corrupt due to the inclusion of non-road pixels in the statistics. It is interesting to note that the mean of the colour spaces that discard luminance information is worse (by about 1 pixel in position and 2 pixels in width) than their counterpart that keep luminance information. This is because of the low colour contrast in the images due to a mostly overcast day.

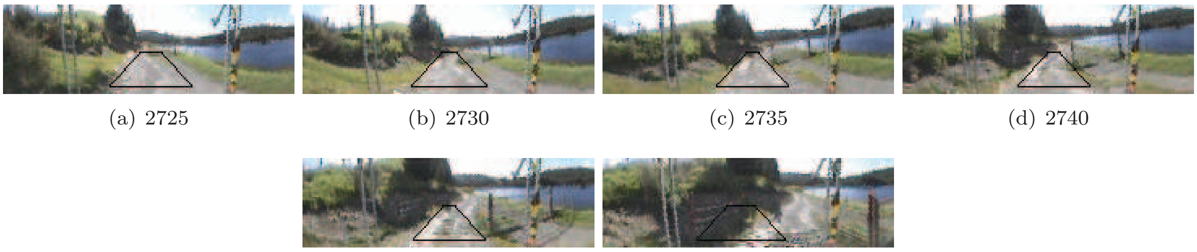
Note that a pedestrian walked in front of the robot (Figure 10(c)) on the road on a small number of frames. This did not affect the performance of the system.

A.2 Cattle grid failure

A few frames captured arriving at the cattle grid and failing are shown in Figure 34. Equivalent frames and beyond the cattle grid are shown in Figure 35.

A.3 Puddle failure

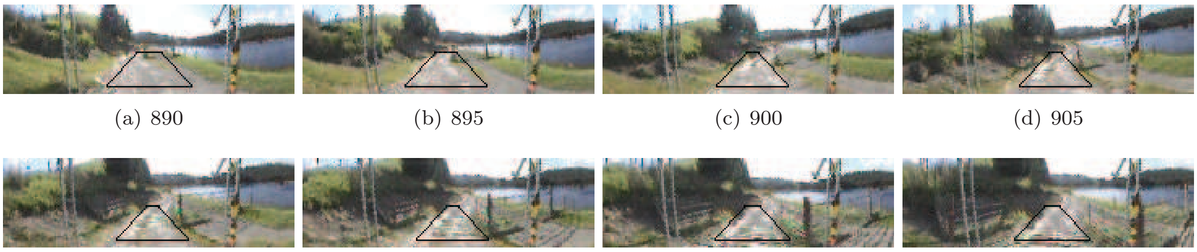
The road leading to the puddle failure goes through a wooded area and is tarmacked but of bad quality and was covered in wet patches and running water. This experiment was performed towards the end of the day, the sun being low and almost aligned with the road when approaching Point 5. The lightness of the grey of the road surrounded by the darkness created by the woods meant that the pixels of the road at the front of



(a) 2725 (b) 2730 (c) 2735 (d) 2740

(e) 2745 (f) 2750

Figure 34: Failure at the cattle grid at Point 4 in Figure 26



(a) 890 (b) 895 (c) 900 (d) 905

(e) 910 (f) 915 (g) 920 (h) 925

Figure 35: Success at the cattle grid at Point 4 in Figure 26

the robot are close to saturation. The large puddle (covering most of the road) arriving in the field of view reflected a dark tree behind it, which was also casting a shadow on that part of the road. This progressively darkened the appearance of the road, the system adapting to the change. The shape of the road at that position (slight turn to the left, the puddle being in the bend, and hump just before the bend) is such that the robot was aligned with a tree just behind the puddle. All this contributed to the robot driving towards the tree and not the road which was still saturated in the images. Some of the images leading to that are shown in Figure 36. This was one of the worst failures but human drivers would have struggled in such a situation too because of the sun more or less aligned with the driving direction. Note that due to the difficult lighting, dust on the tube of the panoramic camera created white saturated spots in the images.

This failure shows that using a forward looking camera would be better because such camera would offer better and more targeted lighting control than the panoramic camera used in this work that tries to perform an overall correct exposure, therefore using non-relevant parts of the scenery.

References

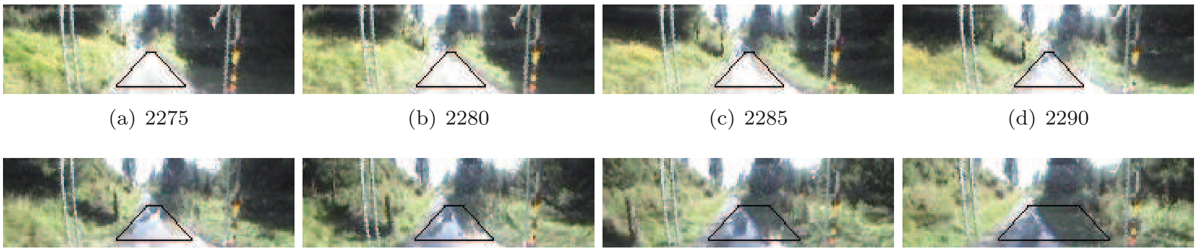
Alvarez, J. M. A. and Ĺopez, A. M. (2011). Road detection based on illuminant invariance. *IEEE Transactions on Intelligent Transportation Systems*, 12:184–193.

Bai, L., Wang, Y., and Fairhurst, M. C. (2008). An extended hyperbola model for road tracking for video-based personal navigation. *Knowledge-Based Systems*, 21(3):265–272.

Bao, J., Chen, Y., and Yu, J. (2012). An optimized discrete neural network in embedded systems for road recognition. *Engineering Applications of Artificial Intelligence*, 25(4):775–782.

Broggi, A. and Cattani, S. (2006). An agent based evolutionary approach to path detection. *Special Issue on Evolutionary Computer Vision and Image Understanding, Pattern Recognition Letters*, 27:1164–1173.

Chen, C.-L. and Tai, C.-L. (2010). Adaptive fuzzy color segmentation with neural network for road detections. *Engineering Applications of Artificial Intelligence*, 23(3):400–410.



(a) 2275 (b) 2280 (c) 2285 (d) 2290

(e) 2295 (f) 2300 (g) 2305 (h) 2310

Figure 36: Detection leading to the puddle failure at point 5 in Figure 26

Crisman, J. D. and Thorpe, C. E. (1988). Color vision for road following. In *SPIE Proceedings on Mobile Robots III*, volume 1007, pages 175–185, Boston, MA, USA.

Crisman, J. D. and Thorpe, C. E. (1991). UNSCARF, a color vision system for the detection of unstructured roads. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 2496–2501, Sacramento, CA, USA.

Crisman, J. D. and Thorpe, C. E. (1993). SCARF: A color vision system that tracks roads and intersections. *IEEE Transactions on Robotics and Automation*, 9(1):49–58.

Dickmanns, E. D. and Mysliwetz, B. D. (1992). Recursive 3-D road and relative ego-state recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):199–213.

Isard, M. and Blake, A. (1998). Condensation—Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 29(1):5–28.

Jeong, H., Oh, Y., Park, J. H., Koo, B. S., and Lee, S. W. (2002). Vision-based adaptive and recursive tracking of unpaved roads. *Pattern Recognition Letters*, 23(1–3):73–82.

Jochem, T. M. and Baluja, S. (1993). A massively parallel road follower. In *Proceedings of Computer Architectures for Machine Perception*, pages 2–12, New Orleans, LA, USA.

Jochem, T. M., Pomerleau, D. A., and Thorpe, C. E. (1995). Vision-based neural network road and intersection detection and traversal. In *Proceedings of the International Conference on Intelligent Robots and Systems*, volume 3, pages 344–349, Pittsburgh, PA, USA.

Katramados, I., Crumpler, S., and Breckon, T. P. (2009). Real-Time Traversable Surface Detection by Colour Space Fusion and Temporal Analysis. In *Proceedings of the International Conference on Computer Vision Systems*, volume 5815 of *Lecture Notes in Computer Science*, pages 265–274, Liège, Belgium.

Kluge, K. and Thorpe, C. (1995). The YARF system for vision-based road following. *Mathematical and Computer Modelling*, 22(4–7):213–233.

Kluge, K. C., Kreucher, C. M., and Lakshmanan, S. (1998). Tracking lane and pavement edges using deformable templates. In *Proceedings of SPIE: Enhanced and Synthetic Vision*, pages 167–176, Orlando, FL, USA.

Labrosse, F. (2006). The visual compass: Performance and limitations of an appearance-based method. *Journal of Field Robotics*, 23(10):913–941.

Land, M. and Horwood, J. (1995). Which parts of the road guide steering? *Nature*, 377:339–340.

Lee, J. and Crane III, C. D. (2006). Road following in an unstructured desert environment based on the em(expectation-maximization) algorithm. In *Proceedings of the International Joint Conference SICE-ICASE*, pages 2969–2974, Busan, Korea.

Li, W., Jiang, X., and Wang, Y. (1998). Road recognition for vision navigation of an autonomous vehicle by fuzzy reasoning. *Fuzzy Sets and Systems*, 93(3):275–280.

Lieb, D., Lookingbill, A., and Thrun, S. (2005). Adaptive road following using self-supervised learning and reverse optical flow. In *Robotics: Science and Systems*, pages 273–280, Cambridge, MA, USA.

Luettel, T., Himmelsbach, M., v. Hundelshausen, F., Manz, M., Mueller, A., and Wuensche, H. J. (2009). Autonomous offroad navigation under poor GPS conditions. In *Proceedings of the 2009 Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, pages 56–62, St. Louis, MO, USA.

Morgan, A. D., Dagless, E. L., Milford, D. J., and Thomas, B. T. (1988). Road edge tracking for robot road following. In *Proceedings of the British Machine Vision Conference*, pages 179–184, Manchester, UK.

Ososinski, M. and Labrosse, F. (2012). Real-time autonomous colour-based following of ill-defined roads. In *Proceedings of Towards Autonomous Robotic Systems*, volume 7429 of *Lecture Notes in Artificial Intelligence*, pages 366–376, Bristol, UK.

Paetzold, F. and Franke, U. (2000). Road recognition in urban environment. *Image and Vision Computing*, 18(5):377–387.

Park, J. W., Lee, J. W., and Jhang, K. Y. (2003). A lane-curve detection based on an LCF. *Pattern Recognition Letters*, 24(14):2301–2313.

Pomerleau, D. A. (1992). *Neural Network Perception for Mobile Robot Guidance*. PhD thesis, School of Computer Science, Carnegie Mellon University.

Procházka, Z. (2013). Road tracking method suitable for both unstructured and structured roads. *International Journal of Advanced Robotic Systems*, 10.

Rasmussen, C. (2004). Grouping dominant orientations for ill-structured road following. In *Proceedings of the IEEE International conference on Computer Vision and Pattern Recognition*, volume 1, pages 470–477, Washington, DC, USA.

Rosenblum, M. (2000). Neurons that know how to drive. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 556–562, Dearborn, MI, USA.

Shinzato, P. Y., Osorio, F. S., and Wolf, D. F. (2011). Visual road recognition based in multiple artificial neural networks. In *Proceedings of the Conferência Brasileira de Sistemas Embarcados Críticos*, volume 1, pages 26–31, São Carlos, Brazil.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C. M., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G. R., Davies, B., Ettinger, S., Kaehler, A., Nefian, A. V., and Mahoney, P. (2006). Stanley: The robot that won the DARPA grand challenge. *Journal of Field Robotics*, 23(9):661–692.

Wang, Y., Teoh, E. K., and Shen, D. (2004). Lane detection and tracking using b-snake. *Image and Vision Computing*, 22(4):269–280.

Woodland, A. and Labrosse, F. (2005). On the separation of luminance from colour in images. In *Proceedings of the International Conference on Vision, Video and Graphics*, Edinburgh, UK.

Zhang, A. M. and Kleeman, L. (2006). A panoramic color vision system for following ill-structured roads. In *Proceedings of the Australasian Conference on Robotics and Automation*, Auckland, NZL.

Zhou, S. and Iagnemma, K. (2010). Self-supervised learning method for unstructured road detection using fuzzy support vector machines. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 1183–1189, Taipei, Taiwan.