Real-time traversable surface detection by colour space fusion and temporal analysis

Ioannis Katramados[1], Steve Crumpler[2] and Toby P. Breckon[1]

[1]Cranfield University, School of Engineering, Cranfield, MK43 0AL, UK
{i.katramados, toby.breckon}@cranfield.ac.uk
[2]TRW Conekt, Stratford Road, Solihull, B90 4GW, UK
steve.crumpler@trw.com

Abstract. We present a real-time approach for traversable surface detection using a low-cost monocular camera mounted on an autonomous vehicle. The proposed methodology extracts colour and texture information from various channels of the HSL, YCbCr and LAB colourspaces by temporal analysis in order to create a "traversability map". On this map lighting and water artifacts are eliminated including shadows, reflections and water prints. Additionally, camera vibration is compensated by temporal filtering leading to robust path edge detection in blurry images. The performance of this approach is extensively evaluated over varying terrain and environmental conditions and the effect of colourspace fusion on the system's precision is analysed. The results show a mean accuracy of 97% over this comprehensive test set.

1 Introduction

This work addresses the problem of autonomous vehicle navigation in semi-structured or unstructured environments where geometrical road models are not applicable. Specifically, a real-time approach is presented which facilitates the detection of traversable surfaces via temporal analysis of multiple image properties. These properties are specifically selected to provide maximally descriptive image information with a minimal computational overhead per image frame. Initially, a multi-stage approach is proposed for feature extraction based on colour and texture analysis. This information is then stored in a temporal memory structure to improve algorithm robustness by means of noise filtering. The proposed methodology has been implemented on the SATURN unmanned ground vehicle as part of the MoD Grand Challenge competition (2008).

Engineering road and obstacle detection systems has long been at the centre of academic and industrial research, leading to a number of successful implementations, ranging from the early road-following systems [4, 16] to the most recent fully automated vehicles in the DARPA Urban Challenge competition (2007) [1, 5, 14]. Additionally, significant research has been motivated by various vehicle platforms for Mars exploration missions [7, 8]. However, the sensing and processing complexity of these systems has often led to costly solutions which whilst useful for exploiting the current limits of technology, do not address the demand for low-cost autonomous platforms utilising widely available low-cost sensors. Creating such vision systems is not a new concept [3, 11]. Embedded lane-departure warning systems [9, 10], are increasingly becoming commonplace in commercial vehicles, motivated by the demand for improved driver safety. However, not every driving environment is as structured as a conventional roadway and an autonomous vehicle may also be required to traverse unstructured environments under varying conditions.

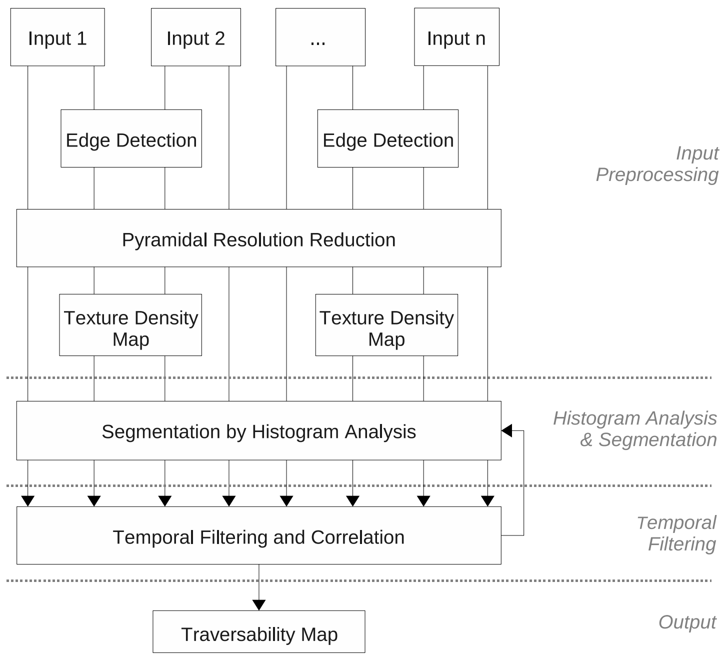


Fig. 1. Traversable area detection methodology

Several prior approaches focus on obstacle detection and avoidance by analysing basic image properties such as texture, colour, hue and saturation of the monocular image. Such approaches are often built on the assumption that the area directly in front of the vehicle is always traversable (initial state assumption) and use a "safe" window to derive the properties of that surface [13]. Obstacles and non-traversable areas are normally identified through a probabilistic model which is based on the similarity of each image pixel to the "safe window" [2, 13, 7]. This becomes the initial *a priori* model from which the system is driven as demonstrated by the Pebbles III robot [13]. The advantages of this approach include flexibility to changing conditions/terrains, limited training requirements and real-time performance. On the other hand, a major disadvantage is its inability to distinguish between surfaces with similar properties due to noise, illumination and environmental effects. To solve this problem Kröse *et al.* [12] proposed the use of optical flow-based techniques, however this is often sensitive to camera vibration and incurs additional computational cost. The work of [12] does however introduce the important aspect of temporal analysis (via frame-to-frame optical flow) as a driver to overcome the earlier limitations of [16, 13]. By contrast, this paper proposes a real-time solution as inspired by the Navlab "Road Following" module [16] and Pebbles III robot [13], with some fundamental changes in the image feature selection from multiple colourspaces and the addition of a novel temporal memory model.

2 Feature extraction for traversable area detection

The following methodology aims to extract information from the video stream output of a vehicle-mounted camera in order to create a map of the traversable and non-traversable areas in real-time. The main challenge is the creation of an algorithm that is adaptable to variable environmental conditions while utilising the least possible computational resource that would facilitate execution on a low-cost processing unit. *Figure 2* provides some examples of such challenging conditions that were experienced during the MoD Grand Challenge competition. The proposed approach is divided into four incremental stages: a) *camera image pre-processing*, b) *multi-dimensional segmentation by histogram analysis*, c) *temporal information processing*, d) *traversable area mapping*. As illustrated in the overview diagram of *Figure 1*, the first stage deals with colour and texture extraction by using intensity-invariant channels of differing colourspaces. The resolution of each input channel is then pyramidically reduced in order to improve system performance and reduce noise (*Figure 1 centre*). Finally, the lower-resolution images are segmented and filtered using a temporal memory model that produces the "traversability" map (*Figure 1 lower*).

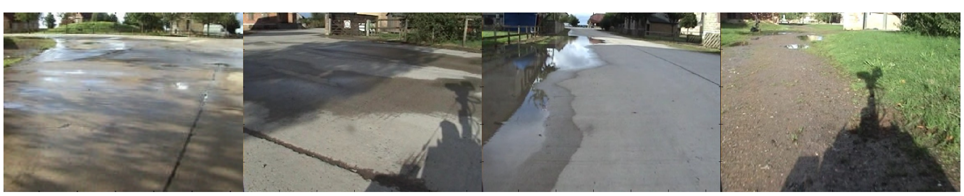


Fig. 2. Examples of challenging environmental conditions with shadows, reflections from standing water and wet prints

2.1 Camera Image Pre-processing

First we describe the noise-filtering approach that is applied prior to segmentation in order to eliminate shadows, reflections and water prints. This is achieved by combining individual channels from differing colourspaces to extract colour and texture information that is insensitive to illumination changes. Prior research [17, 6, 15, 13] has shown that choosing the right colourspace is crucial for extracting accurate path and obstacle features. In fact this methodology combines the *HSL*, *YCbCr* and *LAB* colourspaces [15] to derive four illumination invariant features as listed below:

- **Saturation** (based on the S channel of the *HSL* colourspace)
  By converting the RGB colourspace to *HSL*, the saturation channel is extracted (as illustrated in *Figure 3)* and further resized to a coarse 64×48 saturation intensity map by Gaussian pyramid decomposition of the 320×240 input image.
- **Saturation-based texture**
  This can be derived by applying an edge detector on the S channel of the *HSL* colourspace (*Figure 3*). Then the texture is defined as the density of edges in different parts of the image. Practically, this is achieved by Gaussian pyramid decomposition of the output of the *Sobel* edge detector in order to generate a low-resolution 64×48 grid.
- **Mean Chroma** (based on combining the *Cb* and *Cr* components of the *YCbCr* colourspace with the A component of the LAB colourspace)
  Chroma provides luminance-independent colour information in the *YCbCr* colourspace. As with the S channel of the *HSL* colourspace, dark shadows and reflections alter the chroma level making their detection difficult. To solve this problem Wu et al. [17] propose the combination of the two chroma components (*Cb* and *Cr*) in order to detect features that are entirely light intensity invariant. However, the *Cb* and *Cr* components have a relatively small variation range when compared to the Y component. Based on this observation, the *Cb* and *Cr* values are scaled to fit the $0 - 255$ (8-bit) range and subsequently their mean value is derived. The A channel of the LAB colourspace also provides intensity invariant information, thus by combining it with the mean value of *Cb* and *Cr*, a map of colour distribution (*Figure 4a*) is created as described by *equation 1*.

$$chroma\ map = \frac{sCb + sCr + 2sA}{4} \tag{1}$$

  where sCb is the *Cb* channel of the *YCbCr* colourspace, sCr is the *Cr* channel of the *YCbCr* colourspace and is the A channel of the LAB colourspace. These three parameters have been scaled to 8-bit ($0 - 255$ range).
  An example of a *mean chroma* map is illustrated in *Figure* 4a, where most reflections have successfully been eliminated. This map is also pyramidically reduced to a coarse 64×48 grid.
- **Chroma-based texture** (based on the *Cb* and *Cr* components of the *YCbCr* colourspace)
  This is derived by calculating the mean value of the *Cb* and *Cr* components to generate a new chroma map. The *Sobel* edge detector is subsequently applied to this map in order to calculate a chroma-based texture density

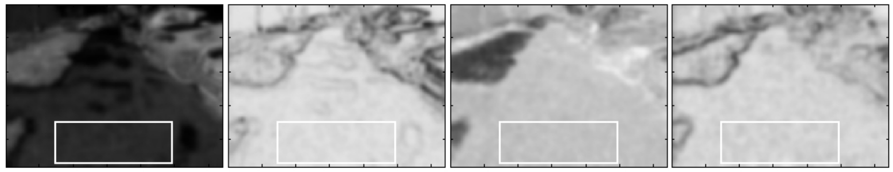


Fig. 3. Image analysis into four input channels: saturation, saturation-based texture, mean chroma and chroma-based texture

using the process described in the saturation-based texture above (*Figure 3*).

At this point we have four 64×48 arrays (8-bit) representing a set of characteristic image properties. These arrays form the input of the segmentation algorithm as described in the following section.

2.2 Segmentation by Histogram Analysis

Several prior path-following techniques have been developed around the assumption that the area immediately in front of the vehicle is initially traversable and thus they identify the pathway by comparison to "safe" window near the bottom of the image [13, 2]. The current approach also adopts this idea since the "safe" window can always be validated by low-cost active short-range sensors such as ultrasonic or infrared. A histogram is calculated for each of the four input image arrays (from the pre-processing stage) within the safe area. The histogram resolution is then reduced by a factor of 8 in order to simplify its processing and improve performance. Thus four different histograms are derived, from which the dominant features of the traversable area are extracted by detecting the histogram peaks based on the assumption that each surface is characterised by a certain combination of saturation, chrominance and texture density levels. Each histogram peak is considered as a feature with five attached properties:

- **Left histogram peak edge**: The point where the left side of the peak meets the "mean level"[1] line
- **Right histogram peak edge**: The point where the right side of the peak meets the "mean level" line
- **Histogram peak value**: The peak value of the low-resolution histogram
- **Mean segment value**: The mean value of the left and right edges of the histogram peak
- **Age**: The time that the peak has remained consistent (in terms of persistence over multiple image frames). A peak is considered as a valid feature only if its age is above a certain threshold. In our tests, the age threshold was set to 10 frames (0.4 sec) with a maximum possible age of 30 frames (1.2 sec).

[1] Defined as the mean of all the histogram values

The left and right histogram peak edges form a histogram segment. Each image pixel is marked as traversable only if its value falls within one of the histogram segments. The remaining pixels are marked as non-traversable. In most cases, the histogram will have only one main segment thus the image will essentially be thresholded. However, more complex surfaces may result in two or more histogram peaks and thus two or more segments. This feature makes the current approach suitable for identifying simple as well as composite traversable surfaces. At this stage, we have four segmented image arrays for each of the four inputs. These arrays are then stored in a temporal memory structure as described in the next section.

2.3 Temporal memory model and correlation

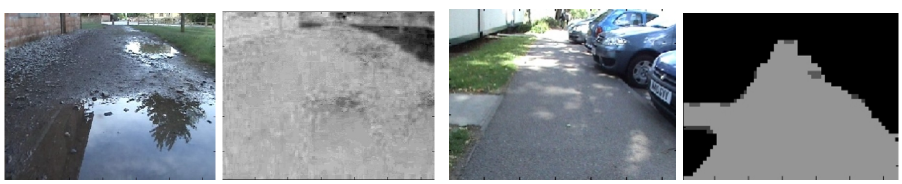
Creating high-level representations of complex raw data can be improved by introducing a temporal memory structure as a way of reducing noise and increasing system accuracy and reliability. This approach proposes the use of temporal behaviour analysis on the output of the segmentation as a top-level filter before correlation. Specifically, the segments identified by histogram analysis are tracked over a series of video frames in order to check their consistency. This is done by assigning a confidence level to each type of surface, which adjusts depending on whether a similar surface appears repeatedly or not. In this way, the system compensates for noise and image blur on a frame-by-frame basis. Similarly, each grid cell of the segmented images is also assigned a confidence level, which increases if its status as "traversable" or "non-traversable" remains unchanged over time. The final output consists of four new "*traversability*" maps based on the saturation, saturation-based texture, mean chroma and chroma-based texture analysis over time. The final traversability map is then derived by majority voting. Although, more sophisticated techniques could have been implemented, this specific one was preferred as the best compromise between overall robustness and real-time performance. Four different levels of traversability are possible for each pixel as illustrated in *Figure* 4*b*, where the darker shades of grey indicate non-traversable areas.

3 Results

The presented approach has been evaluated using a video dataset comprising of sequences captured under a wide range of environmental conditions and different terrain types (*Table 1, Figure* 5). In each video, path and obstacle boundaries (ground truth) were manually marked at *1 sec* intervals. The algorithm output was compared to the ground truth and its accuracy was derived as follows:

$$Accuracy\,(\%) = \left(1 - \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(|g_{ij} - o_{ij}|)}{M \times N}\right) \times 100 \tag{2}$$

where g_{ij} is the ground truth array of size $M \times N$ and o_{ij} is the output array of size $M \times N$. In each of the g_{ij} and o_{ij} arrays the traversable pixels are denoted by '1' and the non-traversable pixels by '0'. Error measurement is then performed by calculating the absolute difference of the two arrays. Note that throughout testing no horizon level was used although this would normally increase the system performance and accuracy further. The results for each scenario are listed in *Table 2*, where the algorithm accuracy was derived using different number of input channels as follows: ***a) 1-channel test*:** Using saturation only, ***b) 2-channel test*:** Using saturation and saturation-based texture, ***c) 3-channel test*:** Using saturation, saturation-based texture and mean chroma and ***d) 4-channel test*:** Using saturation, saturation-based texture, mean chroma and chroma-based texture.



a) *b)*

Fig. 4. *a)* Chroma-based analysis: Areas of low chrominance are eliminated including the foreground water reflections, *b)* Segmentation result after temporal analysis

The algorithm has generally been robust in predicting the traversability of an area regardless of the image quality, noise and camera vibration. *Figure 5* provides some characteristic examples of the system output. As we can see from *Table* 2, a performance of between 95.2% - 97.8% against the ground truth is achieved over a range of conditions (cloudy, wet, sunny, shadow, snow) and a range of terrains (concrete, grass, soil, tarmac, snow) with varying levels of vibration (empirically) recorded on the vehicle platform (*Figures 4b*, 5). The error is measured for each test by calculating the standard deviation of the samples. The overall accuracy and error are then derived by calculating the weighted mean. It should also be noted that using more input channels does not always increase the system accuracy and as a matter of fact the system can sometimes perform better with fewer inputs. This is logical since the colour properties of a surface change with weather and lighting conditions. As a matter of fact, if the system had chosen the right number of input channels for each test, the mean accuracy would have been 97.9% $\pm$ 2.5% (based on the maximum accuracy per test as highlighted by italic characters in *Table 3*). Given the subjective nature of ground truth labelling such a result is also subject to a $\approx$2% error, which is highly acceptable within an autonomous driving scenario.

The evaluation was done using the architecture described in *Figure 1*, which performed in real-time (25 frames per second) when implemented in C++ and executed on a 2GHz Intel Core2Duo CPU using up to four input channels. The camera was mounted on a vehicle that was moving at approximately walking pace. While testing, most obstacles were accurately detected as non-traversable areas except in situations where they were indistinguishable from the underlying surface. Regarding changing environmental conditions (*Table 1*), the performance was good, although reflections were sometimes detected as non-traversable areas. *The video dataset, ground truth data and results can be accessed via the following URL: http://tiny.cc/yannis.*

ID	**Conditions**	**Terrain Type**	**Vibrations**	**Samples**
1	Cloudy Dry	concrete	Light	81
2	Cloudy Wet	concrete	Light	103
3	Cloudy Muddy	soil, grass, gravel	Medium	10
4	Sunny Wet	concrete	Light	20
5	Complex Shadows	tarmac	Very Intense	100
6	Sunny Dry	poor quality tarmac	Very Intense	18
7	Strong shadows	concrete	Light	56
8	Snow	snow-covered tarmac	Medium	104
Total				**492**

Table 1. Environmental and terrain conditions during testing

		1-channel		**2-channel**		**3-channel**		**4-channel**	
ID	*Weight*	*Accuracy*	*Error*	*Accuracy*	*Error*	*Accuracy*	*Error*	*Accuracy*	*Error*
1	0.16	94.87%	3.38%	***97.63%***	1.73%	96.18%	2.20%	97.04%	1.47%
2	0.21	93.68%	4.04%	98.33%	1.09%	98.52%	1.32%	***98.72%***	0.86%
3	0.02	94.35%	2.42%	95.47%	2.96%	97.79%	1.90%	***98.10%***	2.37%
4	0.04	97.12%	4.04%	99.12%	0.74%	***99.42%***	0.24%	99.24%	0.41%
5	0.20	92.47%	8.75%	***96.07%***	5.45%	95.44%	5.50%	95.41%	6.03%
6	0.04	96.46%	3.22%	***98.24%***	1.32%	96.70%	2.81%	96.47%	2.30%
7	0.11	98.95%	1.07%	99.14%	0.73%	***99.20%***	0.66%	99.05%	0.86%
8	0.21	97.01%	4.43%	***98.18%***	3.63%	98.06%	3.04%	98.09%	3.81%
Weighted mean		**95.19%**	**4.57%**	**97.79%**	**2.61%**	**97.44%**	**2.63%**	**97.60%**	**2.70%**

Table 2. Algorithm accuracy results in changing conditions using varying number of input channels (the numbers in bold-italic font denote the test with the highest accuracy in each row)

4 Conclusions

In this paper an effective real-time methodology was presented for detecting traversable surfaces by fusing colour and texture information from *HSL*, *YCbCr* and *LAB* colourspaces to perform image segmentation using a temporal memory model. By initially assuming that the area in front of the vehicle is traversable, the algorithm compares the characteristics of the "safe window" to the rest of the image and creates a "traversability" map. Furthermore, the temporal information is used to filter noise and thus improve system robustness. Testing has proved that this approach is well-suited for autonomous navigation in unstructured or semi-structured environments (up to 97.8% ±2.6% accuracy) and can perform in real-time on platforms with limited processing power. Future work will concentrate on developing an algorithm that can be trained to classify the environmental and terrain conditions in order to optimise colour space fusion.

Acknowledgements

This research has been supported by the Engineering and Physical Sciences Research Council (EPSRC, CASE/CNA/07/85) and TRW Conekt.

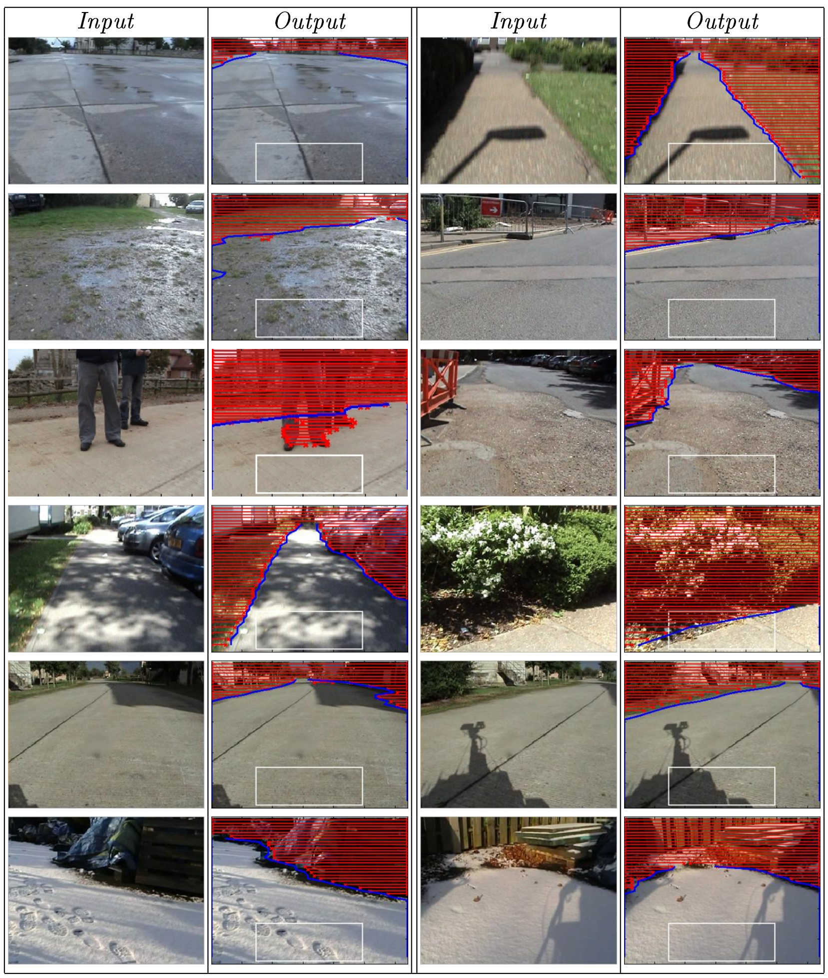


Fig. 5. System input and output examples. Areas covered with red lines are non-traversable. The blue lines indicate the path boundaries and the white box indicates the "*safe window*".

References

1. Y. Alon, A. Ferencz, and A. Shashua. Off-road path following using region classification and geometric projection constraints. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:689–696, 2006.
2. E. M. Amendt and F. W. DePiero. Multi-dimensional k-means image segmentation for off-road autonomous navigation. *10th IASTED International Conference on Signal and Image Processing (SIP)*, 623-065, 2008.
3. P.H. Batavia and S. Singh. Obstacle detection in smooth high curvature terrain. *IEEE International Conference on Robotics and Automation*, 3:3062–3067, 2002.
4. M. Bertozzi, A. Broggi, and A. Fascioli. Vision-based intelligent vehicles: State of the art and perspectives. *Robotics and Autonomous Systems*, 32(1):1–16, July 2000.
5. A. Broggi, C. Caraffi, P.P. Porta, and P. Zani. The single frame stereo vision system for reliable obstacle detection used during the 2005 darpa grand challenge on terramax. *IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 745–752, 2006.
6. R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti. Improving shadow suppression in moving object detection with hsv color information. *IEEE Intelligent Transportation Systems*, pages 334–339, 2001.
7. G.N. Desouza and A.C. Kak. Vision for mobile robot navigation: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, 2002.
8. S.B. Goldberg, M.W. Maimone, and L. Matthies. Stereo vision and rover navigation software for planetary exploration. *IEEE Aerospace Conference Proceedings*, 5:5–2025–5–2036 vol.5, 2002.
9. P. Hsiao and C. Yeh. A portable real-time lane departure warning system based on embedded calculating technique. *IEEE 63rd Vehicular Technology Conference*, 6:2982–2986, 2006.
10. J. Kaszubiak, M. Tornow, R.W. Kuhn, B. Michaelis, and C. Knoeppel. Real-time vehicle and lane detection with embedded hardware. *IEEE Intelligent Vehicles Symposium*, pages 619–624, 2005.
11. K. M. Agrawal Konolige and J. Sola. Large-scale visual odometry for rough terrain. In *International Symposium on Research in Robotics, Hiroshima , Japan*, 2007.
12. B. J. A Krï¿œse, A. Dev, and F. C. A. Groen. Heading direction of a mobile robot from the optical flow. *Image and Vision Computing*, 18(5):415–424, 2000.
13. L.M. Lorigo, R.A. Brooks, and W.E.L. Grimsou. Visually-guided obstacle avoidance in unstructured environments. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1:373–379 vol.1, 1997.
14. A.V. Nefian and G.R. Bradski. Detection of drivable corridors for off-road autonomous navigation. *IEEE International Conference on Image Processing*, pages 3025–3028, 2006.
15. Y. Shan, F. Yang, and R. Wang. Color space selection for moving shadow elimination. *Fourth International Conference on Image and Graphics (ICIG)*, pages 496–501, 2007.
16. C. Thorpe, M.H. Hebert, T. Kanade, and S.A. Shafer. Vision and navigation for the carnegie-mellon navlab. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):362–373, 1988.
17. J. Wu, Z. Yang, J. Wu, and A. Liu. Virtual line group based video vehicle detection algorithm utilizing both luminance and chrominance. *2nd IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 2854–2858, May 2007.