A Decision Network Based Frame-work for Visual Off-Road Path Detection Problem

Alberto Broggi, Claudio Caraffi, Stefano Cattani, Rean Isabella Fedriga
VisLab-Dipartimento di Ingegneria dell'Informazione,
Università di Parma, Parma I-43100, Italy
{broggi,caraffi,cattani,fedriga}@ce.unipr.it

***Abstract*— This paper describes a Decision Network based frame-work used for path-detection algorithm development in autonomous vehicle applications. Lane marker detection algorithms do not work in off-road environments. Off-road trails have too much complexity, with widely varying textures and many differing natural boundaries. The authors have developed a general approach. Images are segmented into regions, based on the homogeneity of some pixel properties and the resulting regions are classified as road or not-road by a Decision Network Process. Combinations of contiguous clusters form the path surface, allowing any arbitrary path to be represented.**

I. INTRODUCTION

On October 8th, 2005 twenty-three vehicles and no drivers gathered in the Mojave desert to compete in the second edition of the DARPA Grand Challenge [1]. The US Defense Advanced Research Programs Administration (DARPA) created this robotic vehicle competition as an open challenge intended to energize the engineering community to tackle the major issues confronting autonomous vehicle development. For the timed competition, DARPA designed a 132-mile off-road desert course that each vehicle had to negotiate. The course was defined by an ordered list of geographic waypoints, a maximum speed for each waypoint and boundaries that could not be crossed. Vehicles had to operate with full autonomy as they maneuvered around obstacles lining the desert course.

This paper presents an artificial vision algorithm developed as a part of the TerraMax™ vehicle. Oshkosh Truck Corporation, Rockwell Collins and the University of Parma partnered together to form Team TerraMax™ [2]. The Team TerraMax™ robotic vehicle (Fig. 1) is a US Marine Medium Tactical Vehicle Replacement (MTVR) truck which was fitted with electronic actuators for steering, brake, throttle, and transmission control. A computer network was installed to host the software applications necessary for autonomous navigation. The applications consisted of vehicle control, path planning, LIDAR obstacle detection, and artificial vision obstacle detection and path following.

The Artificial Vision and Intelligent Systems Lab (VisLab) of the University of Parma developed the artificial vision systems that sensed the environment. Three color cameras captured video images; a single computer processed the data. Obstacle detection used stereo vision and a v-disparity approach. Path detection used monocular images and the approach discussed in this paper.

This work was supported by Oshkosh Truck Corporation

Fig. 1. The TerraMax™ vehicle

II. PATH DETECTION AS A DECISION PROBLEM

Path detection on structured environments has been already successfully faced by the authors [3] and others [4], using monochromatic monocular images and assuming the existence of lane markers. Unfortunately on unstructured and unknown environments is not possible to rely on any *a priori* knowledge about road structure. To overcome this lack of information many different approaches have been proposed: learning the road properties by neural networks [5], selecting the actual road-borders within the set of all possible curves on the image by evolutionary techniques [6], and growing regions believed to belong the road, on the basis of some a priori assumptions or simplifications. All these methods look for a single homogeneous road surface in front of the vehicle, searching based on brightness, color, texture, etc. The hypothesis of uniform homogeneity becomes a huge limitation as it bounds the set of detectable roads to the case of medium/well structured environments.

The proposed method differs from the above methods, overcoming their limitations by generalizing the problem. Roads can have *heterogeneous* surfaces. To find these potentially heterogeneous surfaces, the algorithm looks to build them up from a variable number of smaller *homogeneous* terrain portions. The homogeneous portion of terrain can represent any kind of natural or artificial environment elements, such as gravel or paved roads but also grass, water paddle, oil stains, drivable rocks, lane markers, shadows, etc, and they do not need to be previously learned and recorded in a database [7]. Consequently it is possible to summarize the path detection algorithm as a two-step process:

- divide the image in homogeneous regions made of connected pixels.
- decide which combination of the obtained regions could represent the road surface with the highest probability.

The first step is called clustering, or image segmentation. Clustering in computer vision has been being studied with success, especially for medical applications, using both evolutionary [8] and traditional [9] [10] approaches. However the real-time constraints of the Grand Challenge contest led to the adoption of a simple but fast and easily tunable clustering algorithm, explained in Section III, as a good trade-off between performance and computational requirements.

The second step falls in the class of *decision problems*. Born to help decision processes in medical, transportation, political or environmental fields, *decision theory* [11] is now widely used by Artificial Intelligence researchers as a useful frame-work into which they can map a variety of classical problems. Decision Networks [12] extend Bayesian networks, adding actions and utilities to provide a general methodology for rational decision making. Section IV will describe how Decision Network fits the problem of decide about the set of clusters that belong to the road surface. The decision process tries to minimize the risk of wrong classifications taking into account the current *vehicle state* given by speed, steering angle, steering angle acceleration.

This approach has the advantage of explicitly shifting the path detection to a high level problem, allowing a wider range of situations to be handled in a more sensible way. For example there is no need to remove vehicle shadows at a medium/lower level, on the basis of brightness considerations. In fact since vehicle shadows belong to the terrain surface, its corresponding cluster will be treated just as any other region: like a potential part of the road.

III. CLUSTERING

Several image segmentation algorithms can be found in literature. The most common approach is to join close pixels to obtain a large region composed by similar entities. In approach described in this section, images are decomposed in $d \cdot d$ pixels cells and a comparison is made among them. The use of cells instead of pixels allows a comparison using both the average color value and the information about the texture, like variance.

A. Pseudo Distance Function

To measure the similarity of cells we defined a *pseudo distance function*, that combines distances from cell to cell, from cluster to cluster, and from cell to cluster. Before introducing the distance function it is necessary to define the following entities:

- A set C of $c_{i,j}$, where $c_{i,j} =< x_{k1}, x_{k2}, ...x_{kn} >= c_k \in \Re^n$ is the properties vector of the $k-th$ *cell*, placed in position i, j on the interest area of the image. The interest area is made of the cells that correspond to position not farther that 50m from the vehicle. This information is obtained exploiting the methods described in [13] (See Fig. 3).
- A set V of v_k, where $v_k =< y_{k1}, y_{k2}, ...y_{kn} > \in \Re^n$ is the properties vector of the $k-th$ *cluster* of cells [1].

The partial comparison functions are defined as follows:

- the *cell to cell only* comparison function is:
$$c2c(c_k, c_l) = \sum_{i=0}^{n} D_{i-c2c}(x_{ki}, x_{li}) \cdot \alpha_i, \forall c_k, c_l \in C \quad (1)$$
- the *cluster to cluster only* comparison function is:
$$v2v(v_k, v_l) = \sum_{i=0}^{n} D_{i-v2v}(y_{ki}, y_{li}) \cdot \alpha_i, \forall v_k, v_l \in V \quad (2)$$
- the *cell to cluster only* comparison function is:
$$c2v(c_k, v_l) = \sum_{i=0}^{n} D_{i-c2v}(x_{ki}, y_{li}) \cdot \alpha_i, \forall c_k \in C \text{ and } v_l \in V \quad (3)$$
where $\alpha_i \in \Re^+$ is the fixed weight assigned to the $i-th$ cluster and cell properties, and $D_i(\cdot, \cdot)$ is the corresponding *property comparison function*.

The properties comparison functions must always be greater or equal to zero, but no other constraints are required by our frame-work.

Finally we can define the overall cell to cell *pseudo distance function* $Dist(\cdot, \cdot)$ as follows:

- If c_k,c_l belong, respectively, to the clusters v_k,v_l:
$$Dist(c_k, c_l) = c2c(c_k, c_l) \cdot (1-\beta_c) + v2v(v_k, v_l) \cdot \beta_c \quad (4)$$
- If only c_l belongs to a clusters (v_l):
$$Dist(c_k, c_l) = c2c(c_k, c_l) \cdot (1-\beta_c) + c2v(c_k, v_l) \cdot \beta_c \quad (5)$$
- If only c_k belongs to a clusters (v_k):
$$Dist(c_k, c_l) = c2c(c_k, c_l) \cdot (1-\beta_c) + c2v(c_l, v_k) \cdot \beta_c \quad (6)$$
where $\beta_c \in \Re^+$ is the fixed weight given to the clusters distance when computing the distance between a pair of cells.

Two cells c_k, c_l are *similar* if and only if:
$$Dist(c_k, c_l) \leq q_0 \quad (7)$$
where q_0 is a fixed threshold. Note that $Dist(\cdot, \cdot) \geq 0$, but in general we cannot say nothing about $Dist(a, b) = 0$ and the commutative property. The function $Dist(\cdot, \cdot)$ is applied on cells, but it also takes into account the cluster to which the cells belong, hence the distance two cells with the same characteristics vector could have a distance greater than zero, depending on their clusters. For this reason cells should be selected for comparison based on a *pseudo-random* method, as explained in the next Paragraph.

[1] Hereinafter we will use v_k meaning both the set of cells contained in the $k-th$ cluster and the corresponding properties vector

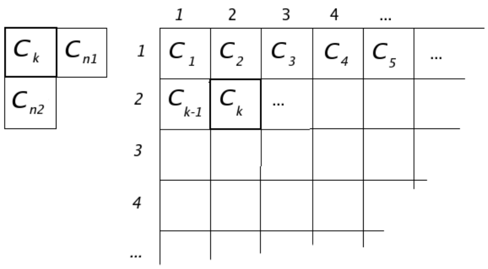


Fig. 2. The 2-neighbors cells c_{n1}, c_{n2}

B. Clustering algorithm

For every pseudo-randomly chosen $c_k \in C$, the clustering process involves the following steps:

1) If c_k does not belong to any cluster a new cluster $v_k = c_k$ is created. This cluster actually corresponds to the c_k cell only.
2) The distances D_{n1}, D_{n2} to the 2-neighbors (see Fig. 2) cells c_{n1}, c_{n2} are computed by (7). Suppose that $D_{n1} < D_{n2}$.
3) When c_{n1} belong to a cluster $v_{n1} \in V$ the following rules, in order of priority, apply:
   - $v2v(v_k, v_{n1}) < q_{c0} \;\; \rightarrow$ the clusters v_k and v_{n1} will be merged, averaging the vectors properties. q_{c0} is a fixed threshold.
   - $(c2v(c_k, v_{n1}) < c2v(c_k, v_k)) \wedge \neg(c2v(c_{n1}, v_k) < c2v(c_{n1}, v_{n1})) \;\rightarrow$ the cell c_k will be moved from v_k to v_{n1}, and the vector properties v_k to v_{n1} will be updated.
   - $\neg(c2v(c_k, v_{n1}) < c2v(c_k, v_k)) \wedge (c2v(c_{n1}, v_k) < c2v(c_{n1}, v_{n1})) \;\rightarrow$ the cell c_n will be moved from v_{n1} to v_k, and the vector properties v_k to v_{n1} will be updated.
   - $(c2v(c_k, v_{n1}) < c2v(c_k, v_k)) \wedge (c2v(c_{n1}, v_k) < c2v(c_{n1}, v_{n1})) \;\rightarrow$ the cells c_k and c_{n1} will be switched, and the vector properties v_k to v_{n1} will be updated.

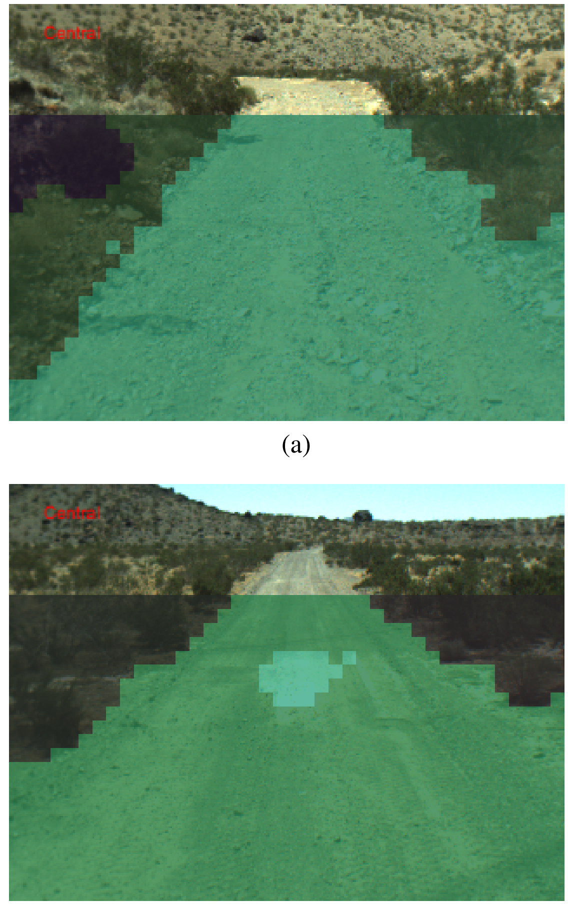
A random selection of the cells helps to avoid the risk of obtaining a final customization that depends on the specific cells comparison order.

At the end of this processing, each cell belongs to one, and only one, cluster in V. Collectively, the clusters will cover the entire interest area of processing. In Fig. 3 cells belonging to the same cluster are drawn with the same color, depending on the cluster's properties vector, hence they appear like one single homogeneous area.

IV. DECISION

The decision phase tries to understand whether a cluster belongs to the road surface, using two different kinds of information:

- clusters properties as a *set of pixels*: homogeneity, size, shape factors.



(a)

(b)

Fig. 3. Clustered images. A stereo based stabilization algorithm is used to move the upper elaboration limit on the image to always corresponding to 50m.

- vehicle state: linear speed, linear acceleration, steering angle, steering angle acceleration.

The underlying idea is the following: each cluster belongs to the road with a given probability depending only on its own *intrinsic* properties: homogeneity (the higher the better), size (the higher the better), shape factors (the closer to road shape the better) and covered freespace area (the wider the better). The probability is computed at each frame without any memory of previous frame classifications. However having high probability of being road is not sufficient, and sometimes even *necessary*, to be finally classified as road. Suppose that the vehicle is running at the highest speed allowed by DARPA in a particular race segment, and is not steering and neither decelerating, but actually is running on the off-road surface. In this situation telling to the path planner that the road is *not* where the vehicle is driving could be extremely risky. This is why sometimes, even if a cluster has a low probability of being a road, the algorithm will classify it as a road and vice-versa. Total risk must be minimized and the risk associated with an incorrect classification depends on the current vehicle state. If the vehicle is running smoothly at high speeds, one should only deliver information about the location of the road which could alter vehicle behavior if one is very sure about the information. In other words, in the decision about a possible classification the following rule applies: classifications with higher risks require higher probabilities (of being correct) before they can be assigned.

A *Maximum Expected Utility* technique includes these risk considerations in the decision process. $U(S)$ denotes the utility of reaching state S as the consequence of a decision D. This utility function assigns a number to express the desirability of a state S. Instead of S, $R_i(D)$ can denote one of the possible outcome of the decision D. Each outcome can occur with a probability $P(R_i(D)|D, E)$, where E summarizes the environment status in which the decision is taken. The *expected utility* of the decision D has the following value:

$$E(D) = \sum_{each R_i(D)} (U(R_i(D)) \cdot P(R_i(D)|D, E)) \tag{8}$$

In other words the expected value of taking a decision is the weighted sum of the possible utility, where the weights are the respective probabilities. A rational decider would take the action with the maximum expected utility.

In the context of the road classification decision, terms in (8) have the following interpretations:

- the possible decisions D are: classify a cluster as road or not road.
- the outcomes R_i are: the classification was correct or not correct.
- the environment E is determined by clusters properties and vehicle state.
- the utility function U depends on the consequences of the corresponding outcome on the vehicle.

Consequently the expected utility value can be expressed as follows:

$$E(road) = P(road) \cdot U_1 + (1 - P(road)) \cdot U_2 \tag{9}$$

$$E(off-road) = (1 - P(road)) \cdot U_3 + P(road) \cdot U_4 \tag{10}$$

where: U_1 is the utility of correctly classifying a cluster as road, U_2 is the utility of incorrectly classifying a cluster as road, U_3 is the utility of correctly classifying a cluster as off-road, U_4 is the utility of a incorrectly classifying a cluster as off-road.

The decision algorithm is implemented as a *cluster growing* process. A cluster c_k, previously classified as road by its E_k, is chosen randomly, then its neighborhood clusters set is analyzed to find a c_j that lead $c_k + c_j$ to an higher likelihood of belonging the road than c_k alone. If such a cluster exist, c_k and c_j fused together and the growing process restart.

The mechanism of cluster growing implements a *one-shot* decision process, where any decision taken at time t does not influence the decisions taken at time $t + 1$, and where decisions are taken only on the basis of the current environment and *current* possible outcomes. The following paragraphs discuss the computation of probability and utility functions.

A. Probability functions

To compute $P(R_i(D)|D, E))$ a complete causal model of the world is required. But having this causal model means having the solution to the path detection problem in general, because we were able to decide accurately when a cluster is actually road or not, by just image processing. Since we have to deal with a completely unknown environment we can only use an *estimation* of this probability distribution.

The probability function has the following form:

$$P(R_i(D)|D, E)) = \sum_{j=0}^{n} (p_j \cdot h_j) \tag{11}$$

where p_j is the probability that would be assigned if one only knew the $j - th$ cluster property (this $j - th$ cluster property is called y_j), with h_j the respective fixed weight. The underlying hypothesis is: the events $R_i(D)$, conditioned to the occurrence of D, y_j, are *disjoint events* when varying y_j. This allows the probability function to be expressed as a linear combination of simpler weighted probability functions.

Moreover the probability function might not involve the same sets of cluster properties when evaluating a cluster alone and when evaluating a set region already classified as road. In Section V we will show an example.

B. Utility functions

The utility functions U_1, U_2, U_3, U_4 give a measure, in $[0, 1] \in \Re$, of the desirability of actions' outcomes. Typically, correct classifications ($U1$,$U3$) have high utilities and incorrect classifications ($U2$,$U4$) have low utilities. However, in some circumstances safe operation might dictate the use of low values of $U3$ and high values of $U2$. These alternate weights increase the likelihood of classifying a given section as a road - even when the estimated probability that the section is, in fact, a road is low. High values of $U2$ (and $U1$) make the utility of classifying the section as a road large. Low values of $U3$ (and $U4$) make the utility of classifying the section as off-road small.

Each outcome has an associated set attributes, v_i, that determine its utility in terms of effect on the vehicle state. The set of attributes is assumed to be *mutually utility-independent (MUI)*[14]: a subset Σ of attributes is utility independent from another subset Γ if the preference between two outcome characterized by Σ_1, Σ_2 does not depend on the values of the corresponding Γ_1, Γ_2. In case of MUI the utility functions can be expressed in the following terms:

$$U = k_1 \cdot U_1 + k_2 \cdot U_2 + k_3 \cdot U_3 + ... + k_1 \cdot k_2 \cdot U_1 \cdot U_2 +$$

$$k_2 \cdot k_3 \cdot U_2 \cdot U_3 + k_1 \cdot k_3 \cdot U_1 \cdot U_3 + ... + k_1 \cdot k_2 \cdot k_3 \cdot U_1 \cdot U_2 \cdot U_3 + ...$$

where U_i is utility function of the the $i - th$ outcome attribute, and k_i the corresponding assigned weight.

C. Decision Network Diagram

Fig. 4 shows the decision network diagram of the framework. The ovals represent *chance nodes*: the random properties over which the decider has no influence (cluster properties). The rectangles represent *decision nodes*: properties

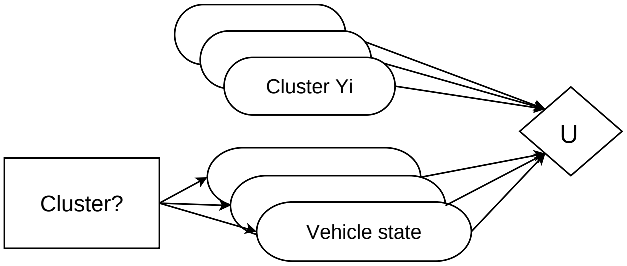


Fig. 4. Decision Network Diagram

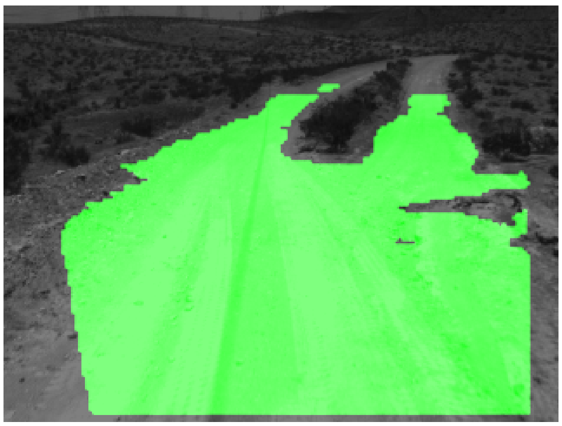


Fig. 5. Freespace obtained by the stereo vision: note that the system detect the whole drivable space in front of the vehicle

influenced by the decider's choice (properties of the vehicle state). The diamonds represent the *utility functions*.

V. IMPLEMENTATION

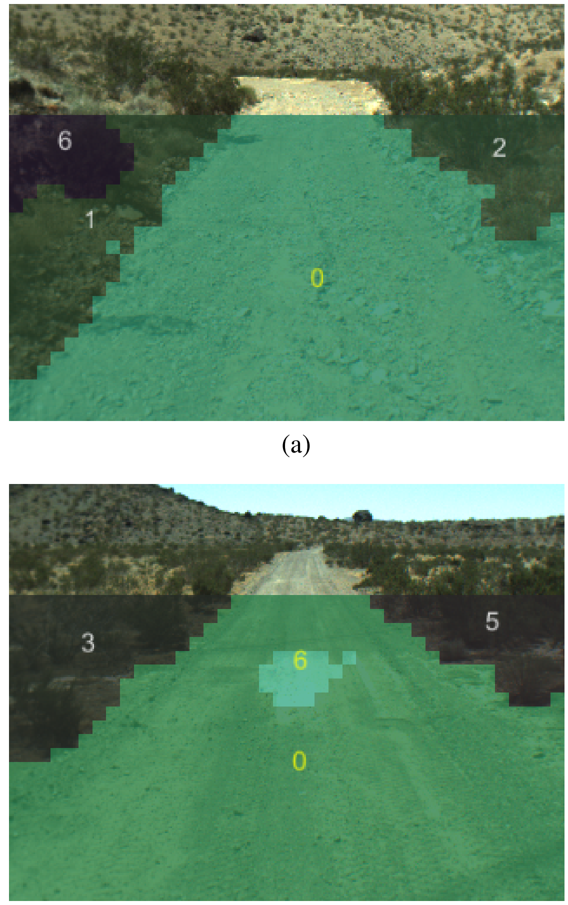
This paper has presented general frame-work, it will now present one specific implementation as an example.

A. Distance Function Implementation

The partial comparison functions showed in (1) (2) (3) are build on the HLS (Hue, Luminance, Saturation) color space. The images are originally acquired in Bayer pattern, and then converted into HLS just before the elaboration. The software acquisition layer also performs a selective gain control correction, maintaining the correct exposure for the obstacle/path detection area for each frame. No other pre-processing is applied on the images.

Cells properties vectors c_k are are built from the respective HLS pixel values (recall that each cell is a $d \cdot d = A$ group of pixels): $<(H_{k1}, \ldots, H_{kA}), (L_{k1}, .., L_A), (S_{k1}, \ldots, S_{kA})> = <(x_{k1}^{[1]}, \ldots, x_{k1}^{[A]}), (x_{k2}^{[1]}, \ldots, x_{k2}^{[A]}), (x_{k3}^{[1]}, \ldots, x_{k3}^{[A]})> = <x_{k1}, x_{k2}, x_{k3}>$. Clusters properties vectors v_k are made of the corresponding cells' average HLS values: $<(\overline{H}_k, \sigma_{Hk}), (\overline{L}_k, \sigma_{Lk}), (\overline{S}_k, \sigma_{Sk})> = <(\mu_{k1}, \sigma_{k1}), (\mu_{k2}, \sigma_{k2}), (\mu_{k3}, \sigma_{k3})> = <y_{k1}, y_{k2}, y_{k3}>$.

Cell to cell (1) partial comparison functions take the following form:



(a)

(b)

Fig. 6. Classification result: (a) the road is made of 2 cluster; (b) the road is made of one single clusters. White numbers denote clusters classified as off-road, yellow as road

$$D_{i-c2c}(x_{ki}, x_{li}) = \frac{\sum_{j=1}^{A} |x_{ki}^{[j]} - x_{li}^{[j]}|}{\sqrt{\sigma_{x_{ki}}^2 + \sigma_{x_{li}}^2}} \tag{12}$$

Cluster to cluster (2) partial comparison functions use cluster averages and variances:

$$D_{i-v2v}(y_{ki}, y_{li}) = \frac{|\mu_{ki} - \mu_{li}|}{\sqrt{\sigma_{ki}^2 + \sigma_{li}^2}} \tag{13}$$

Cluster to vector (3) partial comparison functions also use averages and variances:

$$D_{i-c2v}(c_{ki}, y_{li}) = \frac{|\overline{x_{ki}} - \mu_{li}|}{\sqrt{\sigma_{x_{ki}}^2 + \sigma_{li}^2}} \tag{14}$$

All the above functions are greater or equal than zero. Moreover, (12) and (13) are commutative, and to (13) also applies: $D_{i-c2c}(a, b) = 0 \iff (a = b)$.

B. Clusters properties

The clusters properties y_j used to compute the probability functions p_j in (11) are:

Name	Property
y_1	Average variance of cells' HLS values.
y_2	Number of cells (%).
y_3	Percentage of cells contained in the *freespace*.
y_4	Position and form factors 1.
y_5	Position and form factors 2.

Each property value is normalized to $[0, 1] \in \Re$. The *freespace* is made of the portions of the image assumed to be free of obstacles on the basis of the stereo vision obstacle detector (Fig. 5).

When searching for the first road cluster, the probability function depends only on the first four properties. The fifth property gets included after the initial road cluster gets classified. The refined position and form characterizations in the fifth property can help the cluster growing process to classify small complementary clusters.

C. Vehicle state

The following vehicle state parameters are used to compute the utility values and represent the outcome attributes:

Name	Property
v_1	Current truck speed.
v_2	Current steering angle.

The higher the speed the higher the risk related to change the current path. The farther the computed path is to the current trajectory, the higher the risk related to its classification. Hence the equation in IV-B becomes:

$$U = k_1 \cdot U_{v_1} + k_2 \cdot U_{v_2} + k_1 \cdot k_2 \cdot U_{v_1} \cdot U_{v_2}$$

Fig. 6 shows some classification results. The color of the number super-imposed to clusters represents the classification outcome: white means off-road, yellow means road.

VI. CONCLUSIONS AND FUTURE WORKS

This paper presented a frame-work based on clustering and decision theory to the path detection problem, specifically developed for the DARPA Grand Challenge 2005 and tested on the TerraMax^{TM} autonomous vehicle. The basic idea avoids the trap of starting with a very specific solution, which often works only in very specific conditions, and trying to somehow generalize it to cover all scenarios.

Instead, the approach presented here looks to divide the whole problem into a number of smaller tasks, which have been well studied in literature and for which effective algorithms have been developed. Collectively these tasks can solve the complete problem. This is useful especially in unstructured and very variable environments like the Grand Challenge 2005, where it is almost impossible to rely on any knowledge about the path. Additionally, accounting for the current vehicle state introduces some degree of high-level elaboration into the path detection problem, a typical medium-level task.

The next step will be improving the decision process allowing more complex reasoning. In particular a Markovian Decision Process could improve the decision process by minimizing the risk of converging to a local optimal solution. Moreover a great performance improvement would be given by having the GPS coordinates of waypoints that the vehicle must pass through in the future: in this way the path detector could deal with cross roads where more than one suitable path is visible, deciding which one has to be delivered to the vehicle manager on the basis of the expected trajectory, and also could deal with very sharp curves.

Another improvement could be obtained using a tracking step: a new clusters property, computed on the basis of the predicted most probable road surface at the next step, may be added to the list in V-B and increase the decision process robustness.

The overall algorithm elaboration of a single frame takes about 30ms on a Pentium IV 3.0GHz using 320x240 pixels color images.

VII. ACKNOWLEDGMENT

Authors would like to thank all the members of Team TerraMax^{TM} for the great time spent together developing, testing, and finally seeing our vehicle rolling across the finish line.

REFERENCES

[1] Defence Advanced Research Projects Administration, "Grand Challenge 2005," Avaiable http://www.grandchallenge.org.

[2] Rockwell Collins Oshkosh Truck Corporation and University of Parma, "Team TerraMax 2005," Avaiable http://www.terramax.com.

[3] Massimo Bertozzi and Alberto Broggi, "GOLD: a Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection," *IEEE Trans. on Image Processing*, vol. 7, no. 1, pp. 62–81, Jan. 1998.

[4] R. Aufrère, R. Chapuis, and F. Chausse, "A fast and robust vision based road following algorithm," in *Procs. IEEE Intelligent Vehicles Symposium 2000*, Detroit, USA, Oct. 2000, pp. 192–197.

[5] Christopher Rasmussen, "Combining Laser Range, Color, and Texture Cues for Autonomous Road Following.," in *IEEE International Conference on Robotics and Automation*, Washington, DC, USA, May 2002, pp. 4320–4325.

[6] Alberto Broggi and Stefano Cattani, "An Agent Based Evolutionary Approach to Path Detection for Off-road Vehicle Guidance," *Special Issue on Evolutionary Computer Vision and Image Understanding, Pattern Recognition Letters*, vol. 27, pp. 1164–1173, Aug. 2006.

[7] D. Mateus, G. Avina, and M. Devy, "Robot Visual Navigation in Semi-structured Outdoor Environments," in *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, Apr. 2005, vol. 1, pp. 4691–4696.

[8] N. B. Karayiannis and M. M. Randolph-Gips, "Soft learning vector quantization and clustering algorithms based on non-Euclidean norms: Single-norm algorithms," in *IEEE Transactions on Neural Networks*, 2005, vol. 16, pp. 423–435.

[9] P. H. Batavia and S. Singh, "Obstacle Detection Using Adaptive Color Segmentation and Color Stereo Homography," in *IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 2001, vol. 1, pp. 705–710.

[10] Tae-Kyun Kim and Josef Kittler, "Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 318–327, 2005.

[11] J.Q. Smith, *Decisions Analisys*, Chapman and Hall, London, 1988.

[12] I. Matzkevich and B. Abramson, "Decision analytic networks in artificial intelligence," *Management Science*, vol. 41, pp. 1–22, 1995.

[13] Alberto Broggi, Claudio Caraffi, Pier Paolo Porta, and Paolo Zani, "The Single Frame Stereo Vision System for Reliable Obstacle Detection used during the 2005 Darpa Grand Challenge on TerraMax," in *IEEE ITSC06'*, Toronto, Canada, Sept. 2006.

[14] R.L. Keeney, J.S Hammond, and H. Raiffa, *Smart choices: A guide to making better decisions*, Harvard University Press, Boston, 1999.