# Using Cart to Segment Road Images

**Bob Davies, Rainer Lienhart**

**Angaben zur Veröffentlichung / Publication details:**

Davies, Bob, and Rainer Lienhart. 2005. "Using Cart to Segment Road Images." Augsburg: Universität Augsburg.

# Universität Augsburg

# Using CART to Segment Road Images

**Bob Davies and Rainer Lienhart**

Report 2005-18 Oktober 2005

Institut für Informatik
D-86135 Augsburg

# Using CART to Segment Road Images

Bob Davies[1] and Rainer Lienhart[2]
[1] Intel Research, 2200 Mission College Blvd, Santa Clara, CA 95051
[2] Universität Augsburg, Multimedia Computing Lab, D-86159 Augsburg, Germany

## ABSTRACT

The 2005 DARPA Grand Challenge is a 132 mile race through the desert with autonomous robotic vehicles. Lasers mounted on the car roof provide a map of the road up to 20 meters ahead of the car but the car needs to see further in order to go fast enough to win the race. Computer vision can extend that map of the road ahead but desert road is notoriously similar to the surrounding desert. The CART algorithm (Classification and Regression Trees) provided a machine learning boost to find road while at the same time measuring when that road could not be distinguished from surrounding desert.

**Keywords**: computer vision, image processing, autonomously driving cars, OpenCV, CART, machine learning.

## 1. INTRODUCTION

The DARPA Grand Challenge took place near Primm, Nevada on October 8th, 2005. The GPS (Global Positioning System) waypoints were delivered 2 hours before the race and programmed into the robotic vehicles. The winner of the race had to complete the course in 10 hours or less and with a time better than all the other entrants to win the $2m prize. The complete DARPA rules are found online at http://www.darpa.mil/grandchallenge/. The work for this paper was done in collaboration with the Stanford University Grand Challenge Team (http://www.stanfordracing.org/.) The Stanford team had the best time at 6 hours, 53 minutes, 11 minutes faster than the nearest competitor.

**Figure 1:** Stanford teams' VW Touareg with 5 lasers and a video camera mounted on the roof.

There are a variety of technical challenges involved in an undertaking of this complexity. The purpose of this paper is to focus on how computer vision can reliably contribute to the teams' effort by integrating the vision subsystem with the other sensors on the car.

The core assumption for the vision subsystem is that the car uses a laser subsystem to reliably identify a drivable corridor and the obstacles in the close vicinity of the car. However, the laser system has limitations on how much of the road ahead can be mapped and thus limiting the car's speed. Whenever a new obstacle comes up in the close vicinity or the drivable corridor is changing, the car must be slow enough so that either it can be brought to a complete halt in time in front of the obstacle or can follow a new drivable corridor. The computer vision subsystem receives a projection of the drivable corridor from the laser subsystem and matches similar pixels into the video image. If the additional pixels extend that projection of the drivable corridor, the car can go at a much faster speed.

More expensive lasers with greater range may be mounted on top of the car but as the range increases, the projective geometry makes the precision increasingly dependent on the accuracy of the pose estimation of the car. Any error in the pose estimation increases with the distance from the car. For the purpose of this paper, the lasers are assumed to be accurate only up to 20m ahead of a moving car.

**Software Availability**: The software for this paper is written as a C++ object that is running on Windows XP® and Linux (only Linux is available on the car.) Its portability is enabled through Intel's Open Computer Vision library [2], freely available on the web at SourceForge.net. The source code for this paper is available at http://www.multimedia-computing.de/Code/GrandChallenge/ together with 11 video examples of desert rides and 1000 randomly chosen and manually labeled ground truth images. The labeled images mark the street, the shadow of the car on the street as well as the car hood. This test set is used to evaluate the performance of the proposed methods.

The latest edition of the OpenCV Library contains a number of machine learning algorithms, including Classification and Regression Trees (CART), used for this analysis. In case that this latest release is not yet available at publication time, users will have to replace the two functions for learning and classification by calls to some other machine learning library. The use of the Intel® Performance Primitives library [3] is optional but enhances the performance of the OpenCV library as does the use of Intel's C++ compiler [7]. All these libraries and tools are available on Linux as well as Windows XP.

## 2. BASIC METHODOLOGY

As mentioned in the introduction the car is equipped with lasers in order to predict reliably a drivable, obstacle-free corridor in front of the car. We call this drivable corridor the *laser brick*. However, the laser subsystem can only predict the laser brick as far ahead as the projective geometry is accurate (see Figure 4 for an example of a laser brick). This is not sufficiently far for driving at fast speeds. Therefore a forward-looking camera is mounted on the car, which the vision system will use to extend the road path beyond the projection of the laser brick.

Thus there are two kinds of input sources to the vision system (see Figure 2): (1) the video and (2) the laser brick from the laser subsystem. The laser brick identifies a subset of the actual road pixels in the video. The probability of error is very low. On a set of 1000 randomly selected test images from 11 test videos, the average number of falsely labeled pixels was slightly less than 9%. In other words more than 91% of all pixels in the laser brick actually represent road. This is an error rate most learning algorithms can deal with.

There are two principle ways of learning: discriminant learning or learning generative models. While generative models can explain the observed feature distribution, they usually have a lower classification performance than discriminant learners. Therefore, we made the design choice to use a discriminate learner. As we will see in the experimental results, the performance is significantly higher than a competitive generative model.

All supervised discriminant learning algorithms require labeled positive and negative training examples. Therefore, based on the two input sources the vision system extracts samples of pixels labeled as road and non-road. At this step it is not required to label all pixels, but only those pixels of which we can assume with high probability that our labeling is correct. From these pixels we derive features on which a classifier is trained. For efficiency reasons (fast training at decent classification performance) and its tolerance to outliers CART is used [5], [10]. We will explore in the experimental results section how much performance we could gain using a non real-time learning algorithm such as Boosting [7].

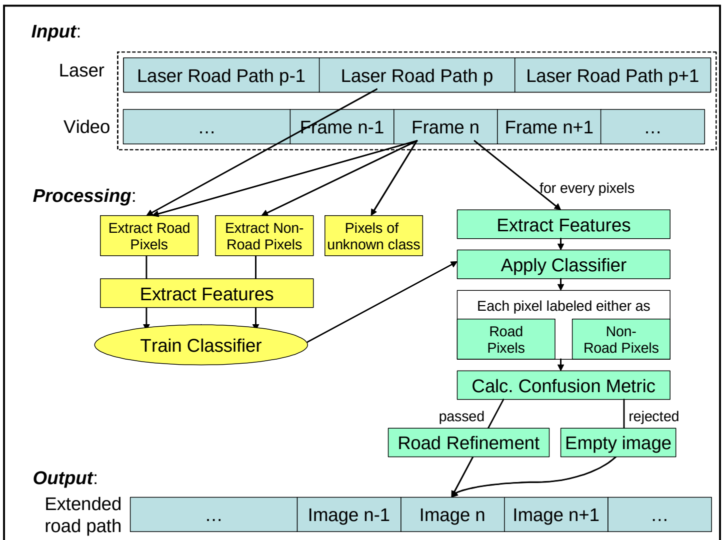


**Figure 2:** Overview of the road segmentation approach.

CART trains on the features of the pixels labeled as road and non-road on a frame by frame basis to build a decision tree that will determine if each pixel in the image is road or non-road. Once the training is complete on a frame, all the pixels in the frame are predicted to be road or non-road using the decision tree that CART produced. A confusion metric will be used to test whether the result seems plausible or must be rejected for inconsistency. If it is plausible, the extrapolated road path or drivable corridor is refined by connected component analysis and reported back to the driving control program or planner. Otherwise an empty image is reported back indicating that the vision system was not able to provide any reliable information.

This overall approach is summarized in Figure 2. We will see that shadow detection will play a crucial role in determining appropriate road and non-road pixels. Therefore, we first turn our focus to the problems that shadows cause and how shadow can be detected.

### 2.1. Shadows

**Shadow Problem:** In image acquisition uneven lighting such as strong shadows under the bright California sun produce severe problems due to the limited dynamic range of CCD sensors. Either the camera's gain control adapts to the bright, shadow-free parts or to the shady parts of the image whichever is larger. In the first case, shadow pixels are dark and show little structure. In the second case, structure is clearly visible in the shaded areas but the bright parts are completely overexposed and all content is blurred. In neither case is the whole visual content of the scene captured; one part will always lack information (see Figure 3).

A practical solution to the shadow problem is to assume that only the laser can see in shadow. This means that if the road is covered in shadow (as in Figure 3), then video cannot provide any benefit and the car must go slow. While this may seem restrictive, most shadow occurs in mountainous terrain where the car must go slow anyway.

**Figure 3:** The road is often obscured by shadow.

There are several ways to enhance the camera's ability to see in shadows such as adding a second camera that overexposes well-lighted areas to gather a good color spectrum in the shadows or possibly clustering separately the pixels that are in shadow and later merging them with the well-lighted regions. However, these alternatives are not considered in this paper. The number of computers on board is limited and only so much capacity can be allocated to vision. Solving the problem of finding road when the road and surroundings are in full daylight is hard enough.

**Defining Shadows**: Estimating what is shadow is pretty straightforward – it means applying a threshold value to a grayscale image. If the image contains a pixel below that threshold, it is considered too dark to have any valuable data and the pixel is added to the mask for shadow. There is a secondary value to knowing where shadow occurs: it defines where land is. One of the more difficult regions to outline and mask off is the sky. It is a safe assumption that there are no shadows in the sky (even dark clouds are well-illuminated) so when shadows are present, the accompanying pixels are likely to be land.

**Figure 4:** The darkened triangle of the area in front of the car is the projection of the laser data into the 2D image from the camera. It is called the laser brick.

One solution to finding the top of the land mass ahead of the car is to scan through the image from top to bottom counting pixels of shadow. As soon as there are a significant number of shadow pixels, the region above the scan contains both sky and horizon land. In steep hills it is possible to find shadows and not have all of the sky – look again at Figure 3 with its steep hillside – but while not all the sky pixels may have been found, many pixels of sky and horizon land will be above the scan line. This horizon land will figure in the estimation of non-road later in the methodology. But there is a further complication with shadow: the area that the laser has defined as road may be in shadow. If this happens, then finding road is impractical because there must be some definition of road that is well-illuminated. The algorithm will have to skip any frame if too little of the laser brick in front of the car is well illuminated.

## 2.2. Defining Road Samples

The lasers mounted on top of the car (see Figure 1) are able to accurately scan the road ahead and build a 3D model for a limited distance. The lasers only scan in a single line on any pass but there are some 180 scans per second. An Inertial Measurement Unit (IMU) is used to compute the pitch and roll of the car while the scan lines are collected. The IMU information and the laser data are integrated and synchronized using carefully calibrated timing to flesh out the 3D model of the road. A description of the use of the lasers can be found in [8], but that reference does not include the calculation of the pitch and roll of the car. More information on the pitch and roll problem may be found at [9].

The laser software produces a 3D map of the road ahead. The precise geometry of the camera mounted on the roof of the car coupled with that 3D laser map enables the laser software to project an approximate outline of the car's expected path into the 2D image produced by the camera. Figure 4 shows what a typical projection of the laser path looks like. Note that the laser often marks as road the flat but rough area next to the road, even including some small plants. Whatever algorithm is used to find road must be robust in light of the fact that the quality of the data may vary for the labeled road and non-road regions. A small amount of noise up to 10% should always been assumed.

The pixels labeled as the laser brick are used to identify road when building the decision tree with CART. Thus the laser brick provides the definition of road for the labeled input to the classification. Once road is defined, the opposite of road – non-road – must then be defined in order to train.

## 2.3. Defining Non-Road Samples

CART requires examples of both road and non-road to build a decision tree that separates road and non-road. So far, there is only a definition of road with the laser brick. There are a multitude of methods to define non-road and this is an area ripe for innovation and experimentation.

The method that was used for this paper and software was to create an initial estimate of non-road using the horizon land. When the shadow mask was built, the first shadows scanning the image from top to bottom define the horizon land. Along the horizon, almost all the land mass is non-road and physically distant from the car. Note that even though the horizon land may be far in the distance, it may still be similar to the road in color and texture. Figure 5 shows an image with the masks for the non-road area as well as the laser brick. The basic shape of an initial estimate of non-road is a rectangle with horizon land at the bottom extending upwards to the top of the image. However, a gap in the horizon land centered on the laser brick is added because the road may extend to the horizon. In addition, the non-road mask is extended at the sides of the image to include non-road areas that are close to the car. The non-road mask should not extend any lower in the image than the top of the laser brick.

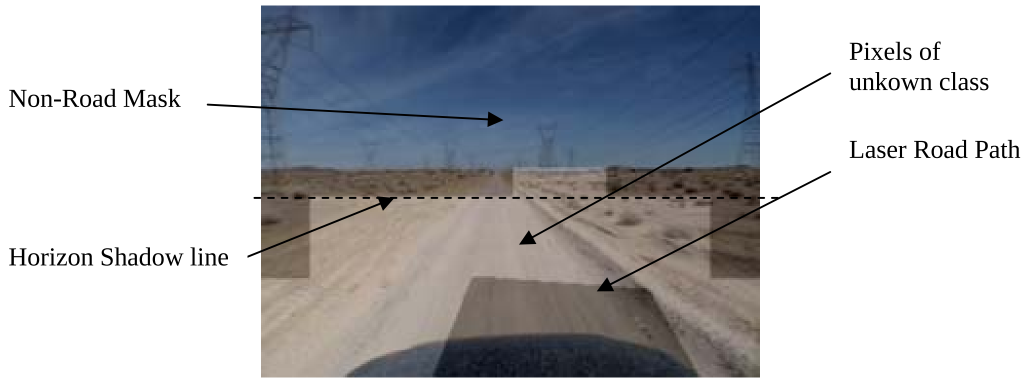


**Figure 5:** The top mask is an initial estimate of non-road while the bottom mask defines road.

Note that the sky is entirely included in the non-road mask. Sky is just another non-road area. Any methodology must also handle images that do not have any sky and one of the benefits of this initial estimate of the non-road area is that there is no dependence on sky being present in the image. The method for finding the horizon land mass finds the sky because it starts including non-road at the top of the image and keeps including horizontal rows of pixels until a significant amount of shadow is found. The initial estimate of the non-road area may include some road. Just as the definition of road from the laser may include some non-road colors and textures, the mask for the non-road will inevitably include some road. The classification learning algorithm needs to allow for some error in both masks.

### 2.4. Confusion Metrics

If the non-road mask includes a lot of road pixels (or hillside pixels that just happen to look like road), the classification will produce results that essentially indicate confusion or ambiguity. The definitions of road and non-road will be too similar to each other. There are 2 measures of this confusion that objectively decide that the image is ambiguous: (1) a lot of road in the laser brick is predicted to be non-road; or (2) a lot of the non-road mask is predicted to be road. If either of these conditions occurs, then the image contains too much ambiguity to segment properly. When ambiguity is detected, the algorithm does not produce a mask to extend the road beyond the laser brick. The thresholds for these measures are set empirically using the videos collected in the desert.

Figure 6 shows a pair of images that demonstrate what confusion looks like. The raw prediction of road pixels on the left shows many pixels that are the same color as the road but are clearly not road. In the event that too much of the non-road mask is predicted to be road, the vision algorithm will enter a confused state where the road is indeterminate. The mapper (a separate subsystem on the car) is collecting all the projections of road, so it is not a problem if no road is identified for a few images. But if there are no recent images that show where the road is, the vehicle must slow down enough so that its stopping distance is less than 20 meters.

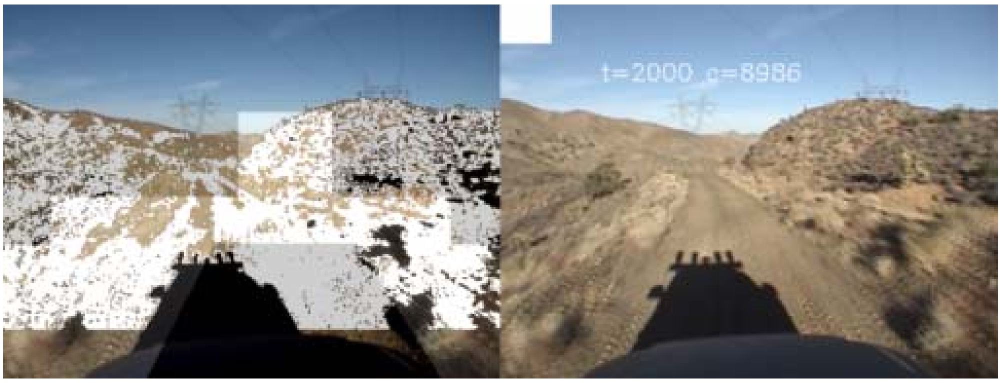


**Figure 6:** The image on the left highlights the raw results of road prediction. There are many pixels similar to the road throughout the image. The image on the right shows the original image where the white box at the top indicates that too many non-road pixels were identified as road and that there is too much ambiguity in the image.

### 2.5. Refining Non-Road Mask

If the previous frame was correctly segmented into road and non-road (without confusion), the prediction for non-road pixels from that previous frame is used as the estimate of non-road in the current frame. The motivation for this is to improve the estimate of non-road. The automated initial estimate of road is comparatively a crude estimate. The initial estimate of the road is only used to begin this cycle of progressively improving and tracking the definition of non-road. All of the work defining the initial estimate of the road is done only until the road is correctly segmented. While there will be differences between frames because of the motion of the car, this definition of non-road is often much better than the initial estimate and, as a result, CART will produce a better decision tree.

### 2.6. Road Refinement

Some conventional computer vision techniques are applied to the results of prediction for each pixel. The reason for this is to eliminate non-road areas that are similar to the road but not connected to the actual road as defined by the laser brick. The results are eroded and then restored with dilation. Then a flood fill is applied to the laser brick so that only road pixels attached to the laser brick are identified as road in the final results.

Figure 7 shows the raw results of prediction after a typical frame has been processed. The frame on the right in Figure 7 shows what happens after the erosion, dilation, and flood fill have taken place. Only those pixels connected to the laser brick are shown as road.


**Figure 7:** The image on the left contains the raw results of prediction. The image on the right contains the identified road after erosion and flood fill have isolated the road pixels.

## 3. CART MODEL AND ITS PERFORMANCE

In Section 2 we have described our overall methodology together with how masks for road and non-road pixels for training are derived. We left out three important aspects:

(1) What features should be extracted for the classifier?
(2) Why did we choose CART over other classifiers?
(3) How do we achieve real-time performance?

In the following we will address these questions.

### 3.1. Features

There are many possible choices with respect to the features that can be derived from the pixels marked as road and non-road. Some examples are the mean and standard deviation for surrounding pixels (of varying sizes and shapes around the pixel of interest), canny edges [15], color correlation vectors [14] or Sobel derivatives (for texture), as well as simply just the red, green, and blue values for the pixel. After experimenting with the different choices, the simplest input – just RGB - produced results visually comparable to those mentioned above while requiring a lot less computation. Therefore RGB pixel colors were chosen as features. However, as we have found in our recent research and indicate in Subsection 3.5, the visual appearance has been misleading, and exploring more complex features may be very rewarding. We are exploring this direction in our current research, and therefore it is beyond the scope of this paper.

### 3.2. CART Road Segmentation Performance

All experiments were performed on a test set of 11 videos of a total duration of 47.5 minutes. The videos were taken in the Mohave Desert on two different days. From these videos 1000 frames were sampled randomly and manually labeled with the ground truth: Street pixels were marked with red, car shadow on the street with green, and the car hood if visible in the video with yellow. The test set with its ground truth can be downloaded at http://www.multimedia-computing.de/ Code/GrandChallenge/.

**Laser Brick**: First we analyzed the precision of the laser brick to understand the bottom line performance. Its road prediction was compared against the marked road in the ground truth (pure red and green pixels). The average recall (= [# of correctly detected road pixels] / [# of road pixels in ground truth]) was about 0.59, while the 1-precision (= false alarm rate = [# of detected road pixels - # of correctly detected road pixels] / [# of detected road pixels]) was about 0.085. In other words, about 59% of all road pixels are marked by the laser brick at an error rate below 9%. At first sight

59% looks high. However, distant road usually consists of only a few pixels, while the nearby road is spatially large in the image. Thus 59% only covers the road pixels very close to the car. In practice, an improvement by 10% in recall from the bottom line 59% of the laser brick often translates into extending the street more than twice as far.

**Drivable Corridor Predicted with CART**: Figure 8 shows the 1-precision (= false alarm rate) vs. recall plot for different training sample sizes. CART-x1 stands for a positive and negative test set with about 600 samples each. Higher numbers such as CART-x3 indicates that the training size is three times higher if possible. On each frame a CART tree was trained. As can be seen from this graph, the best performance values are at about a road recall of 73% and a false alarm rate of 11.9%. On average this translates into 3077 wrongly labelled pixels at 360x240. Out of these 1785 are coming from mistakes in the laser brick. Thus only 1292 wrongly labelled road pixels are added – fewer errors than produced by the laser brick. As we can also see from the graph, the sample set size does not have a large impact on the recall at a false alarm rate below 0.12. This is an important finding since it will allow us to tweak our approach for speed by reducing the sample size for learning.

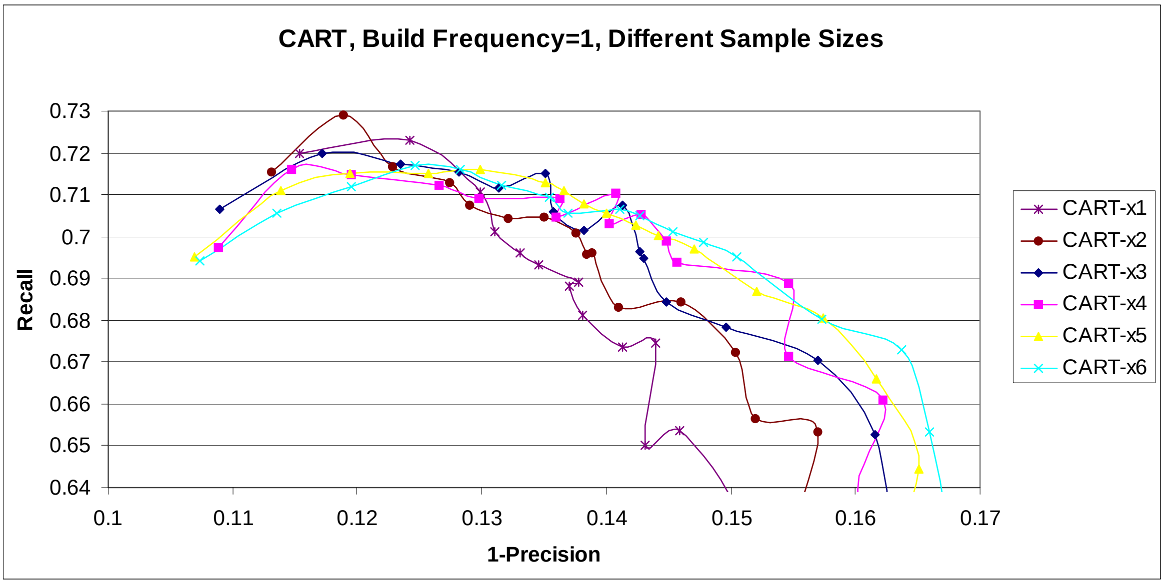


**Figure 8:** Recall vs. 1-Precision (=false alarm rate) plot of road segmentation with CART using different sample size (see 'Sampling' below for more details). CART-x1 stands for a positive and negative test set with about 600 samples each. Higher numbers such as CART-x3 indicates that the training size is three times larger if possible. The CART tree was re-trained every frame. As can be seen from this graph, the best performance values are at about a road recall of 73% and a false alarm rate of 11.9%.

### 3.3. Real-Time Performance

**Sampling:** For each image, CART is trained on the pixels for road and non-road and then CART classifies each pixel in the image. However, there are a lot of duplicate or near duplicate pixels that are input to training. Randomly sampling the input stream of pixels for any one image will produce an almost identical decision tree as one with all the pixels (see Figure 8). There is a limit to how low the number of input pixels to CART may go. The quality of the tree drops off if the sampling rate produces less than 1000 pixels for road and non-road each in total (i.e., less than 500 each).

A chart showing the returns in speed-up from random sampling is given in Figure 9.

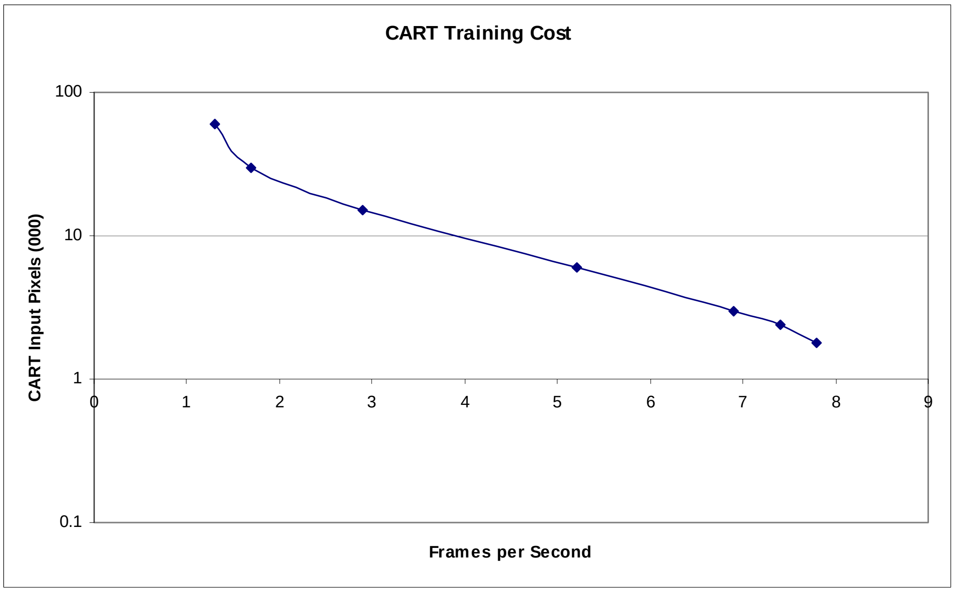


**Figure 9:** The # of frames per second (fps) is dependent on the # of pixels used for training. The more pixels the fewer fps will be achieved. Results are for a Pentium M 1.4 GHz laptop without the Intel Performance Library. Input resolution was 320x240.

**Tree Build Frequency**: It was also investigated how often the decision tree for CART should be rebuilt and what impact on the quality of the results it has. The algorithm in this paper is currently configured to rebuild the decision tree with every frame because colors may vary from frame to frame due to white balance or lighting changes. Rebuilding the decision tree on every frame may be more work than necessary as shown in the following table:

| Build Frequency (BF) | Frame per Second |
|---:|---:|
| 1 | 7.6 |
| 2 | 8.8 |
| 5 | 9.7 |
| 10 | 9.8 |
| 100 | 10.1 |

The segmentation performance for various build frequencies is almost identical as shown in Figure 10. However, for a small increase in the Build Frequency, there is a substantial increase in the frame rate because training is the most expensive part of the algorithm (video processed at 320x240). If a frame is ambiguous because too many road pixels appear in the non-road area, a CART tree is automatically built. Thus the effective build frequency might be higher than the specified one and therefore limits the build frequency going from BF=10 to BF=100.

### 3.4. CART vs. Boosting

We compared the prediction of the drivable corridor with CART against the prediction with GentleBoost – one of the best boosting algorithms [7]. Boosting with decision trees is generally known to be one of the best classification algorithms. However, since multiple weak learners are trained, training time was about 6 times slower and thus beyond our training time budget. As can be seen in Figure 11, a significant improvement could have been achieved if GentleBoost were in our time budget.

This result is important since it indicates that the road / non-road samples contain more information than CART is able to extract from it given the limited CPU resources available in the car. However, as the computational resources go up in future, more sophisticated classification learners can be used to boost road segmentation performance.

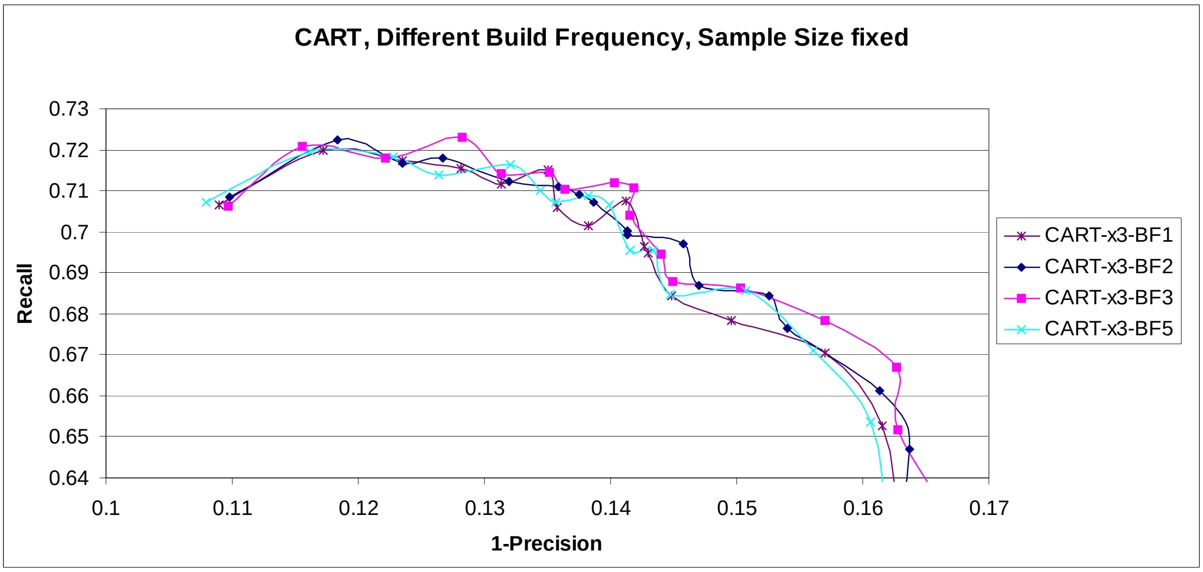


**Figure 10:** There is no need to train a new CART tree every frame. As this graph shows, the Build Frequency (BF) can be significantly reduced without affecting the recall and false alarm rate in the left half of the graph. 'BF*x*' stands for retraining every *x*-th frame or build frequency x.

### 3.5. CART vs. Distribution learner

We also compared our discriminate learner against a generative road model using the same features, i.e., RGB color values (see Figure 12). The generative model is based on stochastic sampling. At every frame the color distribution for road pixels from the laser brick is calculated and pruned by removing rare colors. These colors are than added to a global color distribution estimate based on a simple rule: Colors which occurred frequently in the current road path but are not yet in the global distribution are added to the global distribution estimate with its current probability score. If a color from the current road path already occurred in the global color distribution estimate, its probability is updated by first order exponential smoothing.

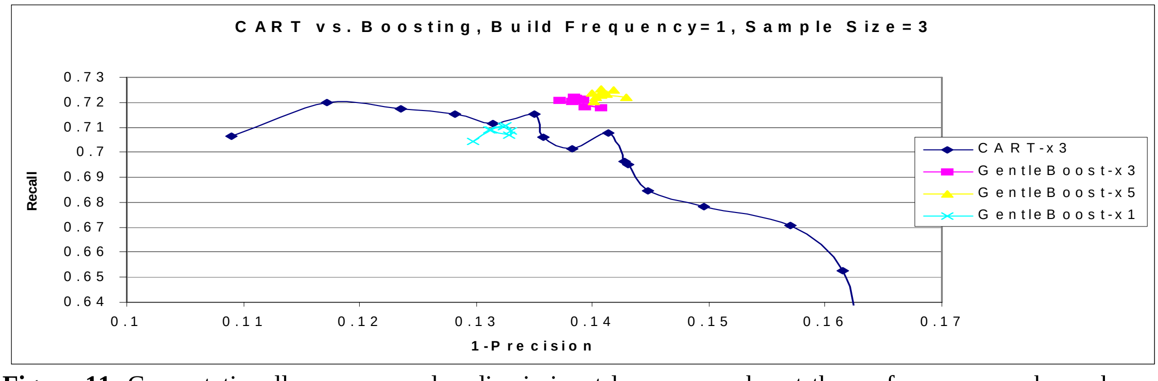


**Figure 11**: Computationally more complex discriminant learner may boost the performance as shown here with GentleBoost.

Classification is performed by labeling all pixels as road pixels that can be explained by the global feature distribution estimate. The road pixels are then filtered (i.e., post-processed) in the same way as described under subsection 2.6 on road refinement.

As can be seen from Figure 12, with the same features (just RGB color values), the generative model labeled DISTEST-Colour in the graph is consistently below the performance of the CART learner at the false alarm rates below 0.125. Three different variants of the color features were used for the generative models by using only the 5, 6, or 7 most significant color bits of each color channel (labeled *_x5, *_x6 and *_x7 in Figure 12).

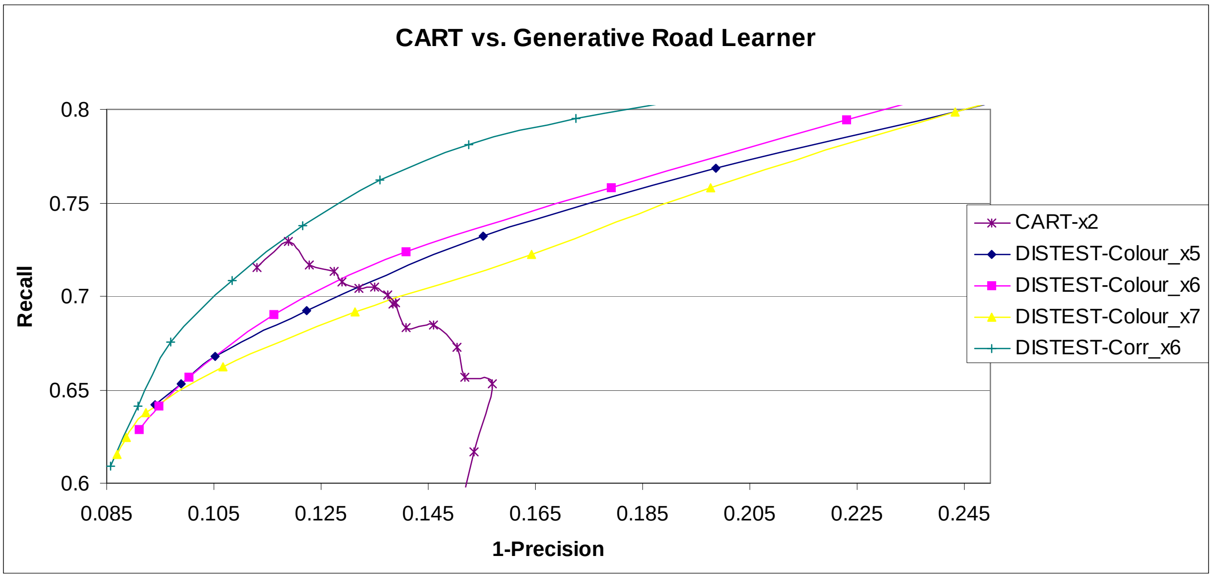


**Figure 12:** Discriminant CART learning and classification (CART-x2) vs. a stochastic and temporal road feature distribution learning (DISTEST-Colour_x#). DISTEST-Corr_x6 shows that a slightly more complex feature can boost the segmentation performance.

The graph, however, also shows the performance for a more complex set of features than just the color value: the color correlation of each pixel with the 5$^{th}$ pixel to its left. The idea behind this feature is to capture that there is a high color correlation between same colors on the road, while this correlation might not exist for the non-road pixels. As can be seen, this slightly more complex feature boosted the performance significantly (see 'DISTEST-Corr_x6; only the 6 most significant bits of each color component involved were taken). Thus, in future work more complex features should also be evaluated with CART learning. We didn't initially investigate this field further because we didn't have ground truth at that time of the decision about the features. Visual inspection might have misled us that it didn't make a difference.

## 4. CONCLUSION

We have shown how monocular vision enhances and extends the laser brick beyond the current 20 meter limit. The approach has been tested against a ground truth and we could experience an improvement in recall from 59% to 73% of road pixels at a low false alarm rate of 11.9% of which 8.5% were cause by errors in the laser brick from the laser subsystem. The CART training is the main contributor to processor overhead but a measured reduction in CART inputs or build frequency can target any desired frame rate. The algorithm has several features that allow a tradeoff between accuracy and speed allowing it to fit into a real-time environment driving an autonomous vehicle and achieve the target 10 frames per second.

## 5. FUTURE WORK

In the future, we plan to investigate stereo cameras for measuring the distance to large objects ahead of the car more thoroughly. Cameras and software from Tyzx [11] and Point Grey [12] were evaluated as part of this effort to find road. More research in the use of these cameras, especially in measuring the error bound for objects in the field of view, could

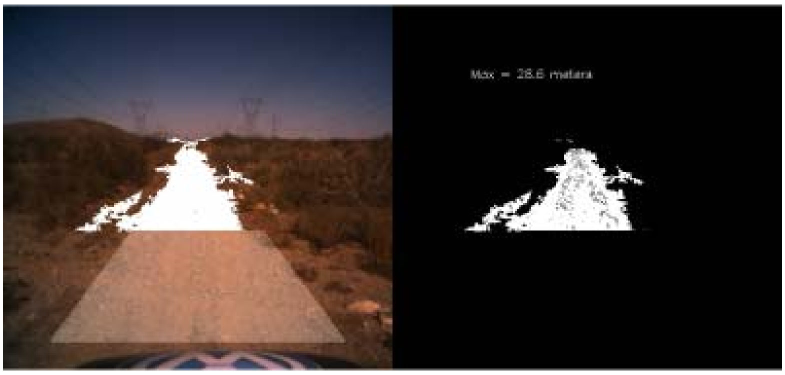


**Figure 13:** Stereo cameras can find distances to large objects. After identifying road in the left image using CART, the mean of the largest distance values produces a reasonable estimate of how far road extends in front of the car. In this example, the road extends over 28 meters ahead of the car.

enhance the ability to detect obstacles. Some experiments with Tyzx showed that it could reliably find the distance to large objects (see Figure 13). For objects that are not as large as the road such as telephone poles or fence posts, the distance values were difficult to narrow down with precision. The distance map might have pixels that are all on the fence post but vary by 10 meters or more when viewed at 50 meters. The baseline of the stereo camera will determine how accurate the results are because of simple triangulation geometry – wider cameras are better. More work could be applied to isolate a better error estimate for use with obstacle detection

Another area for future work is to define non-road in different ways. It remains a rich area for innovation. There are numerous techniques that could be used to find non-road. The better the non-road definition, the better the decision trees for each frame will be. One such technique would be to train and predict twice on every frame, first to predict an estimate of the non-road, then train again with the refined non-road, and predict again for the final result. The cost of this approach was too high for this configuration of the hardware but may be enabled in some future platform. Another method for labeling non-road might define a feature vector for each pixel and use a Euclidean distance from the example pixels of road in the laser brick. This method would define non-road as those pixels whose feature vectors are farthest from the feature vectors for the road – in any of the directions of the n-dimensions of the feature vector.

It should also be noted that only CART and GentleBoost were used to build classifiers for this paper. There are numerous other algorithms such as random forest and SVM that should be tested. The performance constraint eliminated some algorithms for use in the current hardware configuration but the quality of the results can be compared in an offline or batch mode and provide another measure to compare the different algorithms.

Different and more complex features for each pixel value in the image should be explored as well as the use of an infrared camera, which can provide a novel feature vector to enhance the results. The cost of an infrared camera was high enough that no preliminary analysis could be done.

## 6. ACKNOWLEGEMENTS

The authors would like to acknowledge the significant contributions from other members of the vision team working on the Stanford DARPA Grand Challenge robot. Gary Bradski was instrumental in getting Intel involved in the project and numerous lengthy discussions with Hendrik Dalkamp, Adrian Kaelher, David Stavens, Ara Nefian, Vaughn Pratt and Scott Ettinger were invaluable. We also would like to thank Wenyu Que from the University of Augsburg for his hard work on labeling the ground truth images. There is no underestimating the contribution from Sebastian Thrun and Mike Montemerlo, the team leaders for the Stanford robot.

## 7. REFERENCES

1. Open Source Computer Vision Library: http://www.intel.com/research/mrl/research/opencv
2. Intel® Integrated Performance Primitives http://www.intel.com/software/products/perflib

3. Stewart Taylor, “Intel® Integrated Performance Primitives,” in *How to Optimize Software Applications Using Intel® IPP*” http://www.intel.com/intelpress/sum_ipp.htm
4. SourceForge ffdshow encode/decode software, http://sourceforge.net/projects/ffdshow/
5. Breiman L., et. al., Classification and Regression Trees, 1984, ISBN 0-412-04841-8, Chapman & Hall/CRC, New York, NY.
6. Intel C++/C Compiler, http://www.intel.com/cd/software/products/asmo-na/eng/219690.htm
7. Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive Logistic Regression: A Statistical View of Boosting. The Annals of Statistics, 38(2):337-374, April, 2000.
8. Surmann H., Lingemann, K, Nuchter A, Hertzberg J., Fast acquiring and analysis of three dimensional laser range data, http://wwwvis.informatik.uni-stuttgart.de/vmv01/dl/posters/4.pdf
9. Jung, B., Sukhatme, G, Detecting Moving Objects using a Single Camera on a Mobile Robot in an Outdoor Environment, http://wwwvis.informatik.uni-stuttgart.de/vmv01/dl/posters/4.pdf
10. R Project for Statistical Computing, http://www.r-project.org/
11. Tyzx Stereo Cameras, http://www.tyzx.com/
12. Point Grey Stereo Cameras, http://www.ptgrey.com/
13. Jing Huang, S.R. Kumar, M. Mitra, Wei-Jing Zhu; R. Zabih. Image indexing using color correlograms IEEE Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), pp. 762-768, 1997.
14. J. Canny A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, Nov 1986.