

Vision-based Motion Planning for an Autonomous Motorcycle on Ill-Structured Road

Dezhen Song

Dept. of Computer Science
Texas A&M University
College Station, TX 77843

Hyun Nam Lee

Dept. of Electrical Engr.
Texas A&M University
College Station, TX 77843

Jingang Yi

Dept. of Mechanical Engr.
Texas A&M University
College Station, TX 77843

Anthony Levandowski

Unmanned Systems Division
ENSCO Inc.
Falls Church, Virginia 22042

Abstract— We report our development of a vision-based motion planning system for an autonomous motorcycle designed for desert terrain, where uniform road surface and lane markings are not present. The motion planning is based on a vision vector space (V^2 -Space), which is an unitary vector set that represents local collision-free directions in the image coordinate system. V^2 -Space is constructed by extracting the vectors based on the similarity of adjacent pixels, which captures both the color information and the directional information from prior vehicle tire tracks and pedestrian footsteps. We report how V^2 -Space is constructed to reduce the impact of varying lighting conditions in outdoor environments. We also show how V^2 -Space can be used to incorporate vehicle kinematic, dynamic, and time-delay constraints in motion planning to fit the highly dynamic requirements of the motorcycle. The combined algorithm of the V^2 -Space construction and the motion planning runs in $O(n)$ time, where n is the number of pixels in the captured image. Experiments show that our algorithm outputs correct robot motion commands more than 90% of the time.

I. INTRODUCTION

Motivated by the DARPA Grand Challenge¹, we are developing a vision-based motion planning system for an autonomous motorcycle (Fig. 1) to run across a desert terrain or ill-structured roads, where uniform road surface and lane markings do not exist. Since global positioning system (GPS) signals are not enough to guide the vehicle to avoid obstacles and are not always available, additional sensors and decision-making capabilities are needed. Although the single-track platform (motorcycle) provides us with strong off-road capabilities such as excellent agility and navigation on rough terrain, and an ability to pass through narrow openings, its limited size and power supply do not allow us to install sophisticated sensors such as a long distance laser range finder or multiple cameras.

Because of size and power constraints, our motorcycle has one video camera, a GPS receiver, a 3-axis gyroscope, and two on-board computers. The vision system consists of only one camera and one laptop PC while the other computer dedicated to vehicle balance and low level control. Furthermore, the highly dynamic property of the motorcycle demands very responsive vision data processing. These constraints motivate our research to develop a fast and robust vision-based motion planning system for an ill-structured road.

¹<http://www.darpa.mil/grandchallenge/>



Fig. 1. (a) Autonomous motorcycle and (b) an ill-structured road in desert.

We report our development on vision algorithms and systems along with initial experimental results. We propose a concept of the vision vector space (V^2 -Space), which is an unitary vector set that represents local collision-free directions using a 2D image coordinate system. V^2 -Space is constructed by extracting unit vectors based on the similarity of adjacent pixels, which includes the color information and the directional information from prior vehicle tire tracks and pedestrian footsteps. We report how V^2 -Space is constructed using a shadow/illumination invariant color model and a maximum variance color projection method to reduce the impact of varying lighting conditions in outdoor environments. We also show that how V^2 -Space can be used to incorporate vehicle geometric, vehicle dynamic, and time-delay constraints in motion planning to fit the highly dynamic requirements of the motorcycle. The combined algorithm of the V^2 -Space construction and the motion planning runs in $O(n)$ time, where n is the number of pixels in the captured image. Experiments show that it outputs correct robot motion commands more than 90% of the time.

The rest of the paper is organized as follows, we review existing work on vision-based robot motion planning in Section II. We propose V^2 -Space in section III. We present an algorithm for V^2 -Space construction and motion planning in section IV. Experiments are reported in section V and we conclude the paper in section VI.

II. RELATED WORK

Using vision to assist mobile robots and vehicles in navigation has been a popular research field in the past decade [1], [2]. With applications ranging from intelligent vehicles to

autonomous mobile robots, research can be classified based on road conditions, sensing methods, and vision algorithms.

If a robot is running on a *well-structured road*, such as freeways or the roads in an urban area [3], the primary focus of research is lane detection [4] using surface and boundary features, and road following [5], which detects road trends. Since the road has a relatively uniform surface and clear lane markings, techniques such as road segmentation, road edge detection [6], and curve-fitting [7] are often used to generate vehicle control inputs.

When a robot is running in an *unstructured environment* such as a natural environment [8] or the surface of Mars [9], terrain classification and obstacle avoidance become the primary challenges [10]. In such cases, advanced sensors such as stereo cameras, Laser RADAR (LADAR), and appropriate sensor fusion techniques are necessary to deal with the complex environment [11]–[13]. Due to the inherent difficulties in understanding natural objects and changing environments, autonomous driving is still in its infancy. However, existing results such as motion planning with 3D vision and the use of multiple classifiers [10], [14] shed light on a different class of problems, where roads do not disappear completely.

We discuss the motion planning problem for an autonomous motorcycle on a so-called *ill-structured road* [15] that does not have clear lane markings or pavement but might contain the color information and the directional information from prior vehicle tire tracks and pedestrian footsteps. Recent developments in this area are largely driven by the DARPA Grand Challenge. The primary research problem is road identification and obstacle detection. Although the color vision, 3D vision, and the LADAR-based sensing can provide more information for road identification [16]–[20], they cannot be used without significant power and computation capability. Considering the width limitations of the motorcycle platform and the fact that a fast moving vehicle needs to observe the road at a far distance, binocular stereo vision is not very effective due to the “baseline” distance limits between the two cameras [20]. Accordingly, we decided to use monocular vision in our system.

The primary challenge of monocular vision for ill-structured roads arises from several aspects: 1) shadow and illumination changes, 2) no clear road boundaries, 3) drastic changes of road surface, and 4) little or no prior knowledge of the roads. Motion blurring and vibration caused by a fast moving vehicle further degrade image quality. To address these issues, researchers approach the problem using different strategies such as color vision [10], [16], prior knowledge [6], pixel voting [15], classifier fusion [14], optical flow [21], neural networks [3], and machine learning [20], [21].

Raw vision data cannot be directly used to perform motion planning for a robot. For structured environments, high level geometric feature representations such as points, lines, surfaces, and polygons can be used to abstract vision data [22], [23]. For unstructured or ill-structured environments, common feature representations include binary maps and optical flow [21], [24]. The optical flow is the vector field which

warps one image into another (usually very similar) image. The vector field captures both the motion of the robot and other moving objects based on the adjacent video frames. It contains information about moving obstacles and the robot, but it does not work for a still vehicle and is sensitive to motion blurring.

Although desert roads have no clear boundaries, tire tracks and the foot steps left by prior vehicles or pedestrians can provide directional information for vehicle motion planning. Broggi and Berte [4] notice that similar information is provided by lane-markings on urban roads and name it “internal edges”. Rasmussen [15] names the directional information in ill-structured roads as “dominating directions” and uses it to vote for a vanishing point to guide the vehicle. However, one pixel may have more than one dominating direction. A road may fork or intersect with other roads. The directional information can also be trimmed using color information.

The proposed V^2 -Space approach is inspired by the potential field in robot motion planning [25]. Potential field is a virtual Newtonian force field that is proportional to the inverse of the squared distance between the robot and obstacles. A point with bigger force indicates a higher chance of collision. The motion planning is to find the minimum force valley between the robot position and the desired destination. We design V^2 -Space using a similar vector field representation except that vectors here are unitary and only represent local collision-free directions. The V^2 -Space vectors are based on similarity of adjacent pixels within the same frame, and therefore they do not depend on the robot motion and contain possibly multiple directional information. V^2 -Space is designed to capture and represent the information and facilitate motion planning.

III. V^2 -SPACE AND PROBLEM DESCRIPTION

A pin-hole model [27] is used for modeling the on-broad video camera. It is assumed that the camera is calibrated and that both the camera’s intrinsic and extrinsic parameters (with respect to the vehicle) are known. Therefore, we can determine a perspective projection matrix M that projects a point/patch $P = [x y z 1]^T$ in the world frame $\{W\}$ to its corresponding pixel in the image frame $\{I\}$ as $p = [u v 1]^T$

$$[u v 1]^T = M_{3 \times 4} [x y z 1]^T. \quad (1)$$

We assume that the lens distortion of the camera is either negligible or is compensated beforehand.

A. Inputs

Video data from a camera is the primary input for our motion planning system. Define the pixel set of a raw video frame \mathcal{F} with $n = l \times h$ pixels as $I = \{(u, v) | 1 \leq u \leq l, 1 \leq v \leq h, u, v \in \mathbb{N}\}$, where (u, v) are pixel coordinates in $\{I\}$. The video frame \mathcal{F} is a matrix of RGB values

$$\mathcal{F} = (\mathcal{F})_{uv} = ((R, G, B))_{uv}, (u, v) \in I, \quad (2)$$

where $R, G, B \in \mathbb{Z}$ and $0 \leq R, G, B \leq 255$ are integer intensity values for each color channel. Another important input of the motorcycle motion planning system is the direction angle to the next waypoint ϕ obtained from GPS signals.

B. V^2 -Space

V^2 -Space is a collection of unitary vectors that describes local collision-free directions. For frame \mathcal{F} , its V^2 -Space is,

$$\mathcal{V}(\mathcal{F}) = \{\Theta(u, v) : \text{collision-free directions at pixel } (u, v)\}, \quad (3)$$

where $\Theta(u, v) \subseteq [0, 2\pi)$ is a set of collision-free directions at location (u, v) ,

$$\begin{cases} \Theta(u, v) = [0, 2\pi), & \text{If pixel } (u, v) \text{ is on the road} \\ \Theta(u, v) = \emptyset, & \text{If pixel } (u, v) \text{ is an obstacle} \\ \Theta(u, v) \subset [0, 2\pi), & \text{If pixel } (u, v) \text{ is on boundary.} \end{cases} \quad (4)$$

Figure 2 shows an example of the defined collision-free directions at different pixels. Since the V^2 -Space uses the same pixel coordinate of the raw frame in Eq. (2), the perspective projection relationship in Eq. (1) holds between V^2 -Space and $\{W\}$.

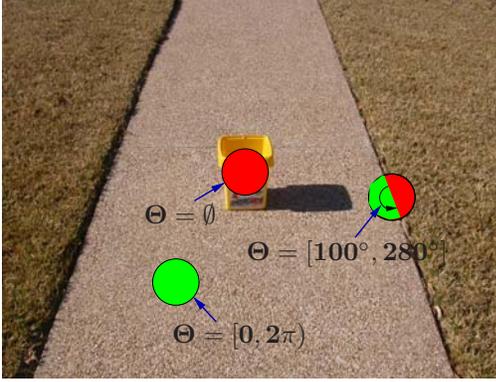


Fig. 2. An example of collision-free directions Θ .

C. Problem Statement

In each iteration, the motion planning system plans for the next τ milliseconds using images \mathcal{F} by outputting a trajectory for subsequent intervals. The inter-iteration time τ_i is strictly less than τ , which allows some overlap in the trajectories between adjacent planning iterations. The planning interval overlap can improve the system robustness and smoothness of the trajectories.

The problem formulation for each planning iteration (with the time period τ_i) is,

Definition 1 (Motion planning): Given \mathcal{F} and ϕ , find trajectory

$$\mathcal{T}_W(\tau) = \{(x(t), y(t)) | t \in [0, \tau)\} \quad (5)$$

for the robot, where $(x(t), y(t))$ is the robot position in $\{W\}$ at time t .

We propose to solve the above motion planning problem in the image frame $\{I\}$. In the following, we first discuss how to compute $\mathcal{T}_W(\tau)$ using the defined V^2 -Space and then present a computational approach to find the optimal motorcycle motion trajectory $\mathcal{T}_W(\tau)$.

IV. ALGORITHMS

We propose to use a computational approach for the motion planning problem. Because the planning period τ_i is small, we can approximate trajectory $\mathcal{T}_W(\tau)$ by a circular curve starting at the current motorcycle position and tangent to the current vehicle velocity. Using such an approximation, we can denote the trajectory $\mathcal{T}_W(\tau)$ by a triplet $(R, d, v^p(t))$ as

$$\mathcal{T}_W(\tau) = \{(R, d, v^p(t)) | R \in [R_{\min}, \infty), d \in \{0, 1\}, t \in [0, \tau)\}, \quad (6)$$

where R is the radius of the trajectory, $d = 0$ (left) or 1 (right) for the trajectory direction with respect to the current velocity, and R_{\min} is the minimal turning radius. We compute $\mathcal{T}_W(\tau)$ given by Eq. (6) in two steps: first, we compute V^2 -Space \mathcal{V} and then we search for a trajectory in \mathcal{V} using a set of circular candidate curves². We begin with the first step of the V^2 -Space construction.

A. V^2 -Space Construction

V^2 -Space construction is a non-trivial feature extraction problem. We propose a three step V^2 -Space construction algorithm as illustrated in Figure 3.

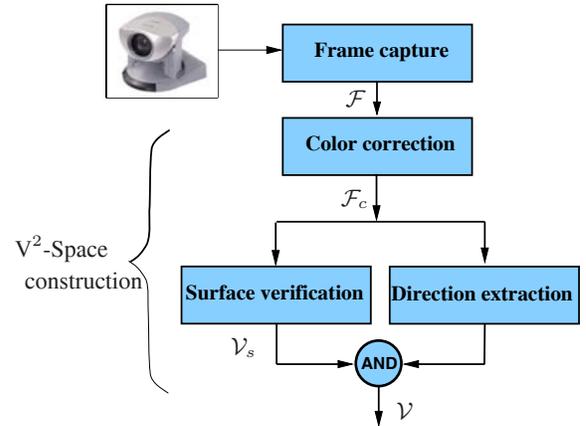


Fig. 3. V^2 -Space construction block diagram.

1) *Color correction:* The purpose of color correction is to minimize the shadow and illumination change effects. Hue, Saturation, and Intensity (HSI) color model has been used widely in road identification research because it is insensitive to illumination [27]. However, our experiments have shown that HSI is not very effective in shadow elimination. We have tested and compared a number of color models such as HSI, normalized RGB, and the $l_1l_2l_3$ and the $c_1c_2c_3$ in [26]. Although the $c_1c_2c_3$ color model is originally designed to be shadow-invariant under the indoor lighting conditions, our experiments show that it is the best shadow and illumination invariant color model for outdoor vision algorithms,

$$\begin{aligned} c_1 &= \arctan\left(\frac{R}{\max(G, B)}\right), & c_2 &= \arctan\left(\frac{G}{\max(R, B)}\right), \\ c_3 &= \arctan\left(\frac{B}{\max(R, G)}\right). \end{aligned} \quad (7)$$

²We will use \mathcal{V} to denote V^2 -Space $\mathcal{V}(\mathcal{F})$.

Figure 4(b) shows that the $c_1c_2c_3$ color model is very effective in shadow elimination. With the corrected color information, our content analysis in subsequent steps becomes significantly more robust. Let us define the output of step 1 as,

$$\mathcal{F}_c = \{(u, v), c_1, c_2, c_3 | (u, v) \in I\}.$$

It takes $O(n)$ time to compute \mathcal{F}_c for an $n = l \times h$ -pixel frame.

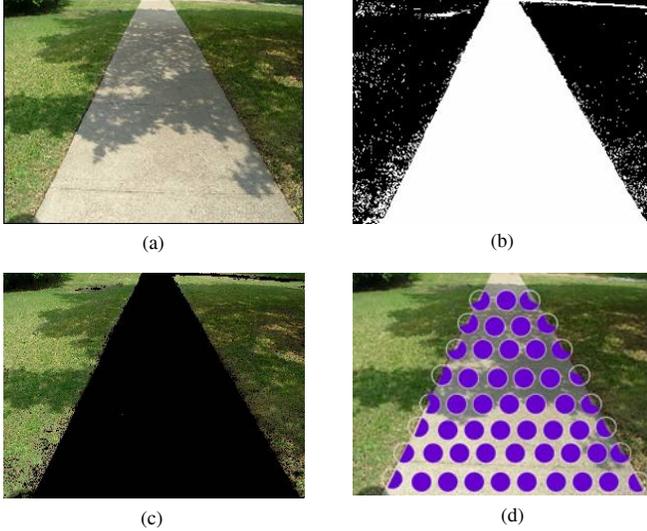


Fig. 4. Collision-free vision space construction. (a) An original video frame with shadow. (b) Classification of the road using shadow invariant color model (c_1, c_2, c_3) (c_3 signature.) (c) The output of surface verification. (d) Collision-free direction information Θ over surface pixels.

2) *Surface verification*: The purpose of surface verification is to identify obstacles and other non-road regions. It outputs a description of free space that the vehicle can pass through. Let us define such free space as \mathcal{V}_s , which takes the same format as Eq. (3). The transformation from \mathcal{F}_c to \mathcal{V}_s is a data-reduction process that builds on both prior knowledge and statistic techniques.

As prior knowledge, we find that desert terrain is not completely unstructured. Vegetation can serve as a nice marking of non-road regions. A large portion of \mathcal{F}_c contains only two types of surface: sandy surface and vegetated surface. A continuously connected surface of the sandy surface is more likely to be a road. Therefore, our first step is to find an effective color discrimination to separate the two types of surfaces. Since vector (c_1, c_2, c_3) is a 3D point in color space, our conjecture is that there should exist an unknown plane in the color space such that the difference between the two types of surfaces is maximized if we project (c_1, c_2, c_3) to the plane. The question then becomes how to find the plane.

Define (w_1, w_2, w_3) as the unitary normal vector of the plane. The color projection of a pixel (u, v) in \mathcal{F}_c is the inner product of two vectors,

$$c_p(u, v) = w_1c_1 + w_2c_2 + w_3c_3, \quad (8)$$

where $c_p(u, v)$ is the separation color that will be used to classify pixels. We employ a data-driven method to estimate (w_1, w_2, w_3) by maximizing the variance,

$$(w_1, w_2, w_3) = \arg \max_{w_1, w_2, w_3} \text{Var}(c_p(u, v)) \\ \text{s.t. } w_1^2 + w_2^2 + w_3^2 = 1$$

To avoid heavy computation load, (w_1, w_2, w_3) does not need to re-computed for each frame. It can be repeated for longer iterations (i.e. every minute).

Now we can reduce \mathcal{F}_c to $\mathcal{F}_p = \{c_p(u, v) | (u, v) \in I\}$. We build on the appearance-based obstacle detection method in [28] to detect obstacles and classify regions. The method is based on the assumption that there exists a reference road region in the image. The reference region is believed to be on the road because it is usually the closest region in front of the robot if the robot stays on the road. The trapezoid region (in $\{I\}$) in Fig. 5(b) is the reference region. Using the pixels in the reference region, we can construct a Gaussian distribution on projected color $c_p(u, v)$. The road surface verification step checks the pixels outside the reference region and classifies them as road or non-road based on the confidence interval constructed from the Gaussian distribution. If pixel (u, v) is located in the confidence interval, then $\Theta(u, v) = [0, 2\pi)$; otherwise, $\Theta(u, v) = \emptyset$. Therefore, it takes $O(n)$ to compute the transformation from \mathcal{F}_c to \mathcal{V}_s . Figure 4(c) shows an example of the surface verification output.

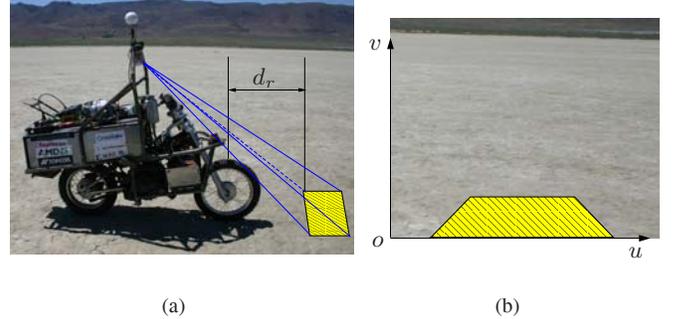


Fig. 5. Appearance-based obstacle detection. (a) Reference region in $\{W\}$. (b) The trapezoid is the reference region in $\{I\}$.

A hidden problem in the method is how to guarantee the reference region is really on the road when the motorcycle is running. Figure 5(a) illustrates a discrepancy d_r between the robot location and the reference region. It causes the planning space to be ahead of the real robot location. Therefore, even if the robot is on the road, the reference region could be outside the road on narrow turns, which can fail the algorithm. We will address this problem later in Section IV-B.3.

3) *Direction extraction*: The purpose of direction extraction is to reduce set \mathcal{V}_s by extracting directional information about the road surface. Although desert roads do not have clear lane markings like structured roads do, they do contain tracks and footsteps left by previous vehicles or pedestrians. These tracks and footsteps can provide useful directional information.

To extract directional information, we must search local collision-free directions for each pixel in \mathcal{V}_s . A straightforward approach is to employ pixel similarity comparison as shown in Fig. 6. Since each pixel has at most 8 neighboring pixels (Fig. 6(b)), we divide $[0, 2\pi)$ into 8 corresponding subsets. We check each direction for pixel similarity. If the neighboring pixel along one direction is statistically similar to pixel at (u, v) , we update $\Theta(u, v)$ accordingly along that direction. Fig. 6(c) illustrated the output $\Theta(u, v)$ for the example.

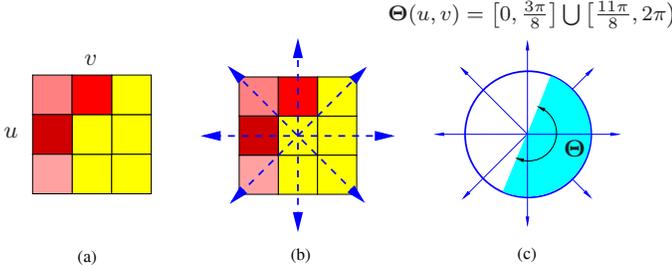


Fig. 6. Directions extracted for a pixel. (a) The pixel at (u, v) . (b) Similarity comparison along eight neighboring directions. (c) Extracted direction information.

To reduce noise effects, in practice we check 5 ~ 10 pixels along each direction. Our approach takes $O(n)$ in this step.

B. Motion Planning in V^2 -Space

With the introduction of \mathcal{V} , the motion planning problem for the motorcycle can be quantitatively formulated. To generate timely and accurate robot control commands, we also need to consider many factors such as image processing delay and motorcycle geometric, kinematic and dynamic limits. We begin with motion planning in V^2 -Space without time delay and a point robot and then consider the factors above to form a complete motion planning solution for the autonomous motorcycle.

1) *A point robot with no time delay*: Using the perspective projection mapping \mathcal{P} by Eq. (1), we can obtain the trajectory \mathcal{T}_I for a set of circular arc trajectory (\mathbf{R}, \mathbf{d}) (bold symbols (\mathbf{R}, \mathbf{d}) denotes a set of (R, d) s in $\{W\}$) as,

$$\mathcal{T}_I = \{(u, v) | (u, v) = \mathcal{P}(\mathbf{R}, \mathbf{d}), \mathcal{P} : \text{projection map}\}. \quad (9)$$

We need to evaluate \mathcal{T}_I in \mathcal{V} to obtain an obstacle-free trajectory. Assuming \mathcal{T}_I overlaps with \mathcal{V} at pixel (u, v) , the direction α at pixel (u, v) of the trajectory is,

$$\alpha(u, v) = \text{atan2}(\Delta u, \Delta v), \quad (u, v) \in \mathcal{T}_I.$$

For a pair of $(R, d) \in (\mathbf{R}, \mathbf{d})$, we can calculate α and then evaluate the trajectory by checking how well it fits in \mathcal{V} . We define a road following quality (RFQ) function $f(u, v; R, d)$,

$$f(u, v; R, d) = \begin{cases} 0, & \text{if } \Theta(u, v) = \emptyset \\ 1, & \text{if } \alpha(u, v) \in \Theta(u, v) \\ |\cos(\theta_d)|, & \text{otherwise,} \end{cases}$$

where $\theta_d = \inf_{\mathcal{V}} |\alpha(u, v) - \partial\Theta|$ is the minimum distance between $\alpha(u, v)$ and $\Theta(u, v)$.

Therefore, we formulate the motion planning problem as an optimization problem: looking for a trajectory that maximizes the RFQ function,

$$\max_{\mathcal{T}_I} \sum_{(u, v) \in \mathcal{V}} f(u, v; R, d). \quad (10)$$

The numerical solution for Eq. (10) will not provide a complete obstacle-free trajectory because along \mathcal{T}_I , $f(u, v; R, d)$ could be zero if one pixel (u, v) is an obstacle. Therefore, we should impose the constraint $f(u, v; R, d) > 0$, $t \in [0, \tau_i)$. The we can find a trajectory given by (R, d) which is obstacle-free.

$$(R, d) = \arg \max_{\mathcal{T}_I} \sum_{(u, v) \in \mathcal{V}} f(u, v; R, d) \\ \text{s.t. } f(u, v; R, d) > 0 \quad (11)$$

We use a set of seven circular candidate $\mathcal{T}_0, \dots, \mathcal{T}_6$ in the implementation. Figure 7 illustrates arcs $\mathcal{T}_0, \dots, \mathcal{T}_6$ in the solution space. Candidate arcs (\mathbf{R}, \mathbf{d}) are defined in the world coordinate system as illustrated in Fig. 7(a) and the projected image in Fig. 7(b).

To choose the velocity profile $v^p(t)$ along the circular trajectory (R, d) , we have to consider several factors. First, the motorcycle cannot run too fast for a given trajectory radius R . If we assume the road surface can provide a constant maximum lateral friction force, for a given turning radius R , the maximum allowable velocity \bar{v} to balance the vehicle has to satisfy $\bar{v}(R) = k_f \sqrt{R}$, where the constant k_f is determined by road/tire interaction properties [29]. On the other hand, we also constrain the motorcycle velocity to be faster than its slowest velocity \underline{v} for stability requirement.

Our current approach is to choose a velocity $v(\tau_i)$ at time τ_i and perform linear interpolation for $v^p(t)$, $t \in [0, \tau_i)$. Recall that $\tau_i < \tau$ is the moment that the next iteration of planning starts. Bounded velocity $v(\tau_i)$ depends on the quality of road ahead,

$$v(\tau_i) = \min\{\underline{v} + \frac{\bar{v}}{S} \sum_{\substack{(u, v) \in \mathcal{V} \\ t \in [\tau_i, \tau)}} f(u, v; R, d), \bar{v}\}, \quad (12)$$

where $S = \sum_{(u, v) \in \mathcal{V}, t \in [0, \tau)} f(u, v; R, d)$. The first term in Eq. (12) calculates the road condition beyond current planning iteration $[0, \tau_i)$ to predict a weighted velocity profile. Larger values of this term indicate better road conditions and hence a faster speed can be achieved.

2) *Incorporating GPS information*: Recall that the GPS input is a direction angle ϕ that points to the next way point. We also need to evaluate each trajectory using ϕ . For \mathcal{T}_I , we have its starting location $(x(0), y(0))$ and the location right before the next iteration $(x(\tau_i), y(\tau_i))$. The overall direction θ_τ in $(0, \tau)$ is,

$$\theta_\tau = \text{atan2}(x(\tau) - x(0), y(\tau) - y(0)).$$

The weight of each trajectory $w(\mathcal{T}_I)$ is based on how much θ and ϕ agree with each other,

$$w(\mathcal{T}_I) = \cos(\theta_\tau - \phi).$$

Therefore, we can use $w(\mathcal{T}_I)$ as a weighted factor of the road following function $f(u, v; R, d)$ in Eq. (11) to calculate the optimal trajectory.

3) *Vehicle size and image processing delay*: Define l_r and w_r as motorcycle length and width, respectively. To guarantee that the reference region in Fig. 5(b) is on the road, we augment the real motorcycle by adding the reference region with discrepancy distance d_r as part of the robot geometric model. Therefore, l_r and w_r are actually larger than the real robot size.

We augment the trajectory evaluation to neighboring regions of the candidate trajectory. Figure 7 illustrates the neighboring region in dashed arcs. For the trajectory in Eq. (6) that starts at $(x(0), y(0))$ in $\{W\}$, the upper envelope of the neighboring region is a concentric arc that starts at $(x(0) + w_r/2, y(0))$ with radius $R + \frac{w_r}{2}$,

$$\mathcal{T}_W^+(\tau) = \left\{ \left(R + \frac{w_r}{2}, d \right) \mid \text{starting at } \left(x(0) + \frac{w_r}{2}, y(0) \right) \right\},$$

and similarly the lower envelope is

$$\mathcal{T}_W^-(\tau) = \left\{ \left(R - \frac{w_r}{2}, d \right) \mid \text{starting at } \left(x(0) - \frac{w_r}{2}, y(0) \right) \right\},$$

With $\mathcal{T}_W^+(\tau)$ and $\mathcal{T}_W^-(\tau)$, we can compute their projection \mathcal{T}_I^+ and \mathcal{T}_I^- by Eq. (9). Define \mathbf{T}_I as the pixels between \mathcal{T}_I^+ and \mathcal{T}_I^- , which is the set of pixels in the neighboring region of \mathcal{T}_I . We can then modify Eq. (10) to incorporate the vehicle size

$$\max_{\mathbf{T}_I} \sum_{(u,v) \in \mathcal{V}} f(u, v; R, d). \quad (13)$$

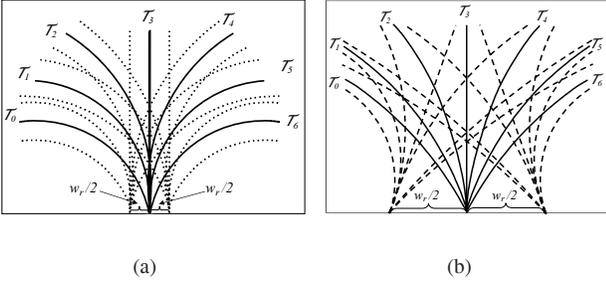


Fig. 7. Sample candidate arc trajectories and vehicle boundaries in (a) world coordinate system $\{W\}$ and (b) image coordinate system $\{I\}$. The solid arcs are candidate trajectories while dashed arcs are augmented boundaries of the vehicle that characterize the size of the vehicle and the size/location of the reference region defined in Fig. 5. Each solid arc has two corresponding dashed arcs.

Image capturing, processing, communication, and the robot control all take time and these actions result a time delay. Such a delay can be further classified as measurement delay and decision/execution delay. Measurement delay t_m refers to the elapsed time from the moment that the camera captures a frame to the moment that the RGB data enters computer memory. Decision/execution delay t_d refers to the interval between the moment that the system takes the frame from memory to the moment the robot actually executes the resulting control command from the algorithm output.

Assuming $t = 0$ at the beginning of each iteration, motion planning is then based on the frame captured t_m time ago and the command generated will be executed t_d time later. To address such a time discrepancy, we can compensate for the time delay by shifting the starting location of the planned trajectory to its actual location at t_d . Figure 8 illustrates how to compensate for the delay. Without loss of generality, we assume that $\{W\}$ has its origin at the center of the lower edge of the camera field of view. Then the last known position with respect to \mathcal{V} is $(x(-t_m), y(-t_m)) = (0, -d_r)$, where d_r is the discrepancy distance illustrated in Fig. 5. Therefore, we can estimate $(x(t_d), y(t_d))$ by taking velocity integrals over time period between $-t_m$ and t_d . The estimated $((x(t_d), y(t_d)))$ is the new starting point of trajectory $\mathcal{T}_W(\tau)$.

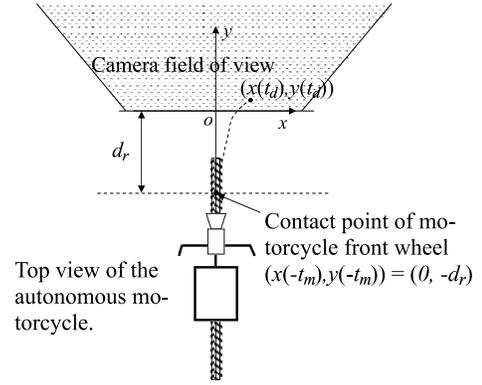


Fig. 8. A schematic of delay compensation.

C. Algorithm

Combining the analysis above, we have motion planning Algorithm 1. It is clear that the overall algorithm runs in $O(n)$ time.

Algorithm 1: Motion planning algorithm

input : Monocular vision image and GPS information

output: Collision-free trajectory $\mathcal{T}_W(\tau)$

Construct the collision-free vector space \mathcal{V}

Vision processing delay compensation

Generate circular path candidates and its projection \mathcal{T}_I

for each trajectory $T_i \in \mathcal{T}_I$ **do**

 Compute the GPS weighting factor $w(T_i)$

 Augment the robot size and compute \mathbf{T}_I

 Compute the objective function

$$F_i = \sum_{(u,v) \in \mathcal{V}} w(T_i) f(u, v; \mathbf{T}_I)$$

$$\mathcal{T}_W(\tau) = \arg \max_{T_i \in \mathcal{T}_I} F_i$$

 Generate velocity profile $v^p(t)$ with dynamics constraints

V. EXPERIMENTS AND RESULTS

We have implemented the algorithm on a laptop PC with a 1.6 GHz Centrino processor and 512 Mb RAM. The camera used is a Canon VCC3 camera with a 47.5° horizontal field of view. Based on Microsoft Direct X SDK version 9.0, our

algorithm can run with an input from either a live video from the camera or pre-recorded video clips. Our algorithm can process the video at a speed of 5 frames per second.

A. Experiments with Video Clip

The first step of the experiments is to test the algorithm using the video data from the route of DARPA Grand Challenge. Figure 9 illustrates the algorithm using one of the snapshots in the two hour video clip. Figure 9(a) illustrates the results of surface verification. Black pixels represent regions that look close to the road surface. It is clear that the data is very noisy because the difference between the road and its surrounding environment is not significant. However, after directional information is extracted, the resulting \mathcal{V} in Fig. 9(b) is quite a good fit of the real road (we use the circular direction to indicate the direction information at each pixel). Even though part of the road is very bright and misclassified as obstacles, the overall \mathcal{V} is sufficient for motion planning. Table I and Fig. 9(c) illustrates the result of candidate arcs evaluation without GPS inputs. The vision algorithm ranks three top choices including arcs 3, 4, and 5. Figure 9(d) show how a GPS signal is used to identify the final choice. Figure 10 uses two more examples to further illustrate how a GPS signal can be used to improve the quality of the output of the vision algorithm. Note that the starting points of the arcs in all examples are calculated with considering vehicle kinematic constraints and time-delays.

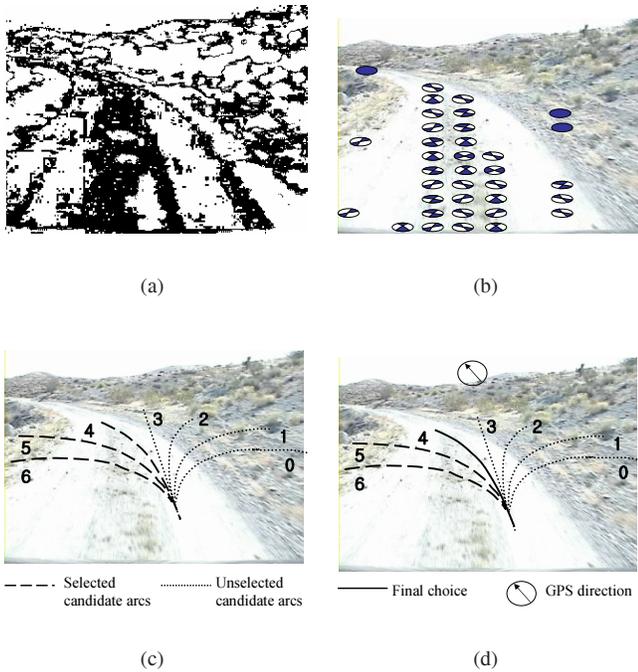


Fig. 9. An illustration of V^2 -Space algorithm using a snapshot of the video clip captured in the Mojave desert. (a) \mathcal{V}_s , (b) \mathcal{V} , (c) result of the arc evaluation using Eq. (13), and (d) final choice of the arc with GPS inputs.

During the test, we found that the algorithm has a successful classification rate of 91%. The failure cases tend to happen

TABLE I
CANDIDATE ARC EVALUATION OF ROAD FOLLOWING FUNCTIONS USING
EQ. (13) FOR THE EXAMPLES IN FIGS. 9 AND 10

Fig.	0	1	2	3	4	5	6
9	271.2	129.6	88.0	147.3	717.1	512.2	286.4
10(a)	556.7	524.0	1079.0	1303.8	1330.3	475.4	321.3
10(b)	794.4	960.7	1346.8	1031.3	922.2	1105.6	794.8

at the moment when the road surface change drastically. For example, if part of the road is covered by water, the algorithm cannot distinguish the water from obstacles. Another problem is caused by the inherent limitations of monocular vision. If the surface of the road is identical to the surface of an obstacle, the algorithm fails because it cannot tell the difference.

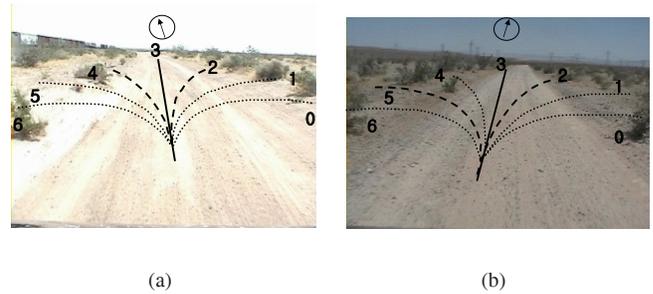


Fig. 10. Two examples using the video data from the Mojave desert.

B. Field Tests

Before we test the algorithm on a running motorcycle, we ran it on a smaller mobile robot. The smaller robot is a three wheel robot with two front driving wheels and one rear caster as illustrated in Fig. 11(a). The robot is 30 cm wide and 45 cm tall and can travel at a speed of 25 cm per second with 25 lbs payload. It is also equipped with two wheel encoders and digital compass.

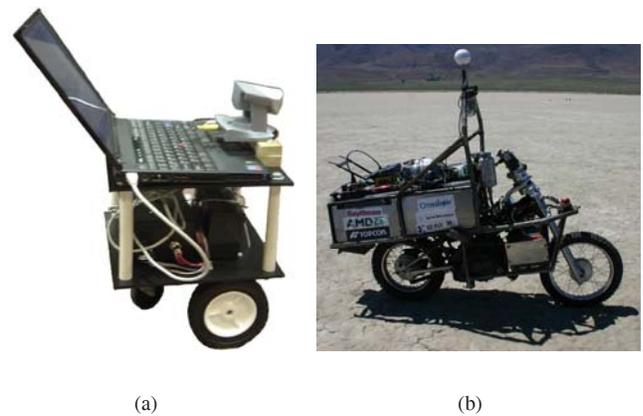


Fig. 11. Robots used for field tests.

We conduct experiments in a golf course, local parks, and on the university campus. The robot can follow the road correctly

92% of the time, which is better than the video clip results because the road conditions are less difficult than that of the video clip. As illustrated in Fig. 11(b), the most recent testing motorcycle is built on a 100 cc Yamaha bike. With a customized drive-by-wire system, this vehicle can drive at a speed of up to 60km/h. We have built three vehicles for testing. We are currently working on testing the bike in the field and will report more results in future versions of this paper.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we report our development of a vision-based motion planning algorithm for an autonomous motorcycle. To efficiently process video data and perform motion planning, we propose V²-Space, a new framework that represents road features and allows fast construction and motion planning. We use a shadow and illumination invariant color model to construct V²-Space to reduce the impact of varying lighting conditions in an outdoor environment. We extract directional information from prior tire tracks and pedestrian footsteps on the road to refine our V²-Space. The V²-Space also allows us to consider vehicle kinematic, dynamic, and time-delays in motion planning to fit the highly dynamic requirement of the motorcycle. We propose a V²-Space construction and motion planning algorithm that runs linear to the number of pixels. The algorithm is tested both with video clips from the desert and in field experiments. It outputs correct robot motion commands at a successful rate of more than 90%.

More experimental testing on an autonomous motorcycle is currently under the way. In the future, we will consider incorporating V²-Space in a stereo vision system. We will perform partial construction of the real 3D environment in V²-Space to allow fast computations. We will also incorporate machine learning techniques into V²-Space to improve the vehicle's capability of adapting to different terrains.

ACKNOWLEDGMENTS

Thanks are given to N. Qin, Q. Hu, J. Koh, E. Grant, A. Parish for implementing part of the project. Our thanks to K. Goldberg, D. Volz, R. Gutierrez-Osuna, V. Taylor, and N. Amato for insightful discussions and feedback.

REFERENCES

- [1] G. N. Desouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 2, pp. 237–267, 2002.
- [2] M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta, "Artificial vision in road vehicles," *Proc. IEEE*, vol. 90, no. 7, pp. 1258–1271, 2002.
- [3] R. Aufriere, J. Gowdy, C. Mertz, C. Thorpe, C.-C. Wang, and T. Yata, "Perception for collision avoidance and autonomous driving," *Mechatronics*, vol. 13, no. 10, pp. 1149–1161, 2003.
- [4] A. Broggi and S. Berte, "Vision-based road detection in automotive systems: A real-time expectation-driven approach," *J. of Artificial Intell. Res.*, vol. 3, pp. 325–348, 1995.
- [5] M. Bertozzi and A. Broggi, "Gold: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Trans. Image Processing*, vol. 7, no. 1, pp. 62–81, 1998.
- [6] M. Ekinci, F. W. J. Gibbs, and B. T. Thomas, "Knowledge-based navigation for autonomous road vehicles," *Turkish J. of Elec. Eng. & Comp. Sci.*, vol. 8, no. 1, pp. 1–29, 2000.
- [7] Y. He, H. Wang, and B. Zhang, "Color-based road detection in urban traffic scenes," *IEEE Trans. Intell. Transport. Syst.*, vol. 5, no. 4, pp. 309–318, 2004.
- [8] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," *Autonomous Robots*, vol. 18, no. 1, pp. 81–102, 2005.
- [9] R. Volpe, T. Estlin, S. Laubach, C. Olson, and J. Balaram, "Enhanced mars rover navigation techniques," in *Proc. 2000 IEEE Int. Conf. Robotics and Automation*, San Francisco, CA, April 2000, pp. 926–931.
- [10] L. M. Lorigo, R. A. Brooks, and W. E. L. Grimsou, "Visually-guided obstacle avoidance in unstructured environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Grenoble, France, September 1997, pp. 373–379.
- [11] M. J. Stephens, R. J. Blissett, D. Charnley, E. P. Sparks, and J. M. Pike, "Outdoor vehicle navigation using passive 3d vision," in *Proc. IEEE Comp. Soc. Conf. Comp. Vision and Pattern Recog.*, San Diego, CA, June 1989, pp. 556–562.
- [12] L. Matthies, T. Litwin, K. Owens, A. Rankin, K. Murphy, D. Coorobs, J. Gilsinn, T. Hong, S. Legowik, M. Nashman, and B. Yoshimi, "Performance evaluation of UGV obstacle detection with ccd/flir stereo vision and lidar," in *Proc. IEEE ISIC/CIRA/ISAS Joint Conf.*, Gaithersburg, MD, September 1998, pp. 658–670.
- [13] J. Ibanez-Guzman, X. Jian, A. Malcolm, Z. Gong, C. Chan, and A. Tay, "Autonomous armoured logistics carrier for natural environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Sendai, Japan, September 2004, pp. 473–478.
- [14] C. Dima, N. Vandapel, and M. Hebert, "Classifier fusion for outdoor obstacle detection," in *Proc. 2004 IEEE Int. Conf. Robotics and Automation*, New Orleans, LA, April 2004, pp. 665 – 671.
- [15] C. Rasmussen, "Grouping dominant orientations for ill-structured road following," in *Proc. IEEE Comp. Soc. Conf. Comp. Vis. and Pattern Recog.*, Washington, DC, June 2004, pp. 470–477.
- [16] S. Buluswar and B. Draper, "Color machine vision for autonomous vehicles," *Int. J. for Eng. Appl. of Artificial Intell.*, vol. 1, no. 2, pp. 245–256, 1998.
- [17] J. Crisman and C. Thorpe, "Unscarf, a color vision system for the detection of unstructured roads," in *Proc. 1991 IEEE Int. Conf. Robotics and Automation*, Sacramento, CA, April 1991, pp. 2496 – 2501.
- [18] M. Kolesnik, G. Paar, A. Bauer, and M. Ulm, "Algorithmic solution for autonomous vision-based off-road navigation," in *Proc. SPIE: Enhanced and Synthetic Vis.*, Orlando, FL, vol. 3364, April 1998, pp. 230–247.
- [19] T. Hong, C. Rasmussen, T. Chang, and M. Shneider, "Road detection and tracking for autonomous mobile robots," in *Proc. SPIE 16th Ann. Int. Symp. Aero./Def. Sensing, Simul., and Contr.*, Orlando, FL, April 2002.
- [20] J. Michels, A. Saxena, and A. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *Proc. 22nd Int. Conf. Machine Learning*, Bonn, Germany, August 2005.
- [21] D. Lieb, A. Lookingbill, and S. Thrun, "Adaptive road following using self-supervised learning and reverse optical flow," in *Proc. Robotics: Sci. and Syst.*, Cambridge, MA, June 2005.
- [22] H. Zhang and J. P. Ostrowski, "Visual motion planning for mobile robots," *IEEE Trans. Robot. Automat.*, vol. 18, no. 2, pp. 199–208, 2002.
- [23] N. J. Cowan, J. D. Weingarten, and D. E. Koditschek, "Visual servoing via navigation functions," *IEEE Trans. Robot. Automat.*, vol. 18, no. 4, pp. 521–533, 2002.
- [24] J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi, "Divergent stereo for robot navigation: learning from bees," in *Proc. IEEE Comp. Soc. Conf. on Comp. Vis. and Pattern Recog.*, New York, NY, June 1993, pp. 434–439.
- [25] Y. Hwang and N. Ahuja, "A potential field approach to path planning," *IEEE Trans. Robot. Automat.*, vol. 8, no. 1, pp. 23–32, 1992.
- [26] T. Gevers and A. W. M. Smeulders, "Colour based object recognition," *Pattern Recognition*, no. 32, pp. 453–464, 1999.
- [27] D. Forsyth and J. Fonce, *Computer Vision: A Modern Approach*, Prentice Hall, Upper Saddle River, NJ, 2003.
- [28] I. Ulrich and I. Nourbakhsh, "Appearance-based obstacle detection with monocular color vision," in *Proc. AAAI Nat. Conf. on Artificial Intell.*, Austin, TX, 2000.
- [29] J. Yi, D. Song, A. Levandowski, and S. Jayasuriya, "Trajectory tracking and balance stabilization control for autonomous motorcycles," in *Proc. 2006 IEEE Int. Conf. Robotics and Automation*, Orlando, FL, May 2006, pp. 2583–2589.