

Visual Servoing for Mobile Ground Navigation

Jeffrey Kane Johnson

Maeve Automation

Pittsburgh, PA USA

contact@maeveautomation.org

Abstract—This paper presents a vision-based control framework that attempts to mitigate several shortcomings of current approaches to mobile navigation, including the requirement for detailed 3D maps. The framework defines potential fields in image space and uses a subsumption process to combine hard, physical constraints with soft, guidance constraints while guaranteeing that hard constraint information is preserved. In addition, this representation can be defined with constant size, which can enable strong run-time guarantees to be made for visual servoing-based control. The framework is demonstrated with proof-of-concept examples in simulation and the real world, as well as data sets and an open source implementation.

Index Terms—visual servoing, subsumption, navigation

I. INTRODUCTION

This paper presents a mobile navigation control framework intended to mitigate three problems often encountered in current approaches to ground navigation. First, many navigation systems require detailed 3D maps, but these maps are often either difficult or expensive to create and maintain or unavailable [1], [2]. Second, agents often encounter large and varying numbers of entities in a scene while navigating, which can be problematic for many popular planning and control approaches whose complexities are sensitive to entity counts [3], [4]. Third, many robotic systems, especially safety critical ones, have strict resource and running time requirements [5] and may be subject to rigorous verification and validation procedures [6] that are difficult to perform with current approaches. To help address these issues, this paper expands on [7] to present a subsumption control framework built around Image Space Potential (ISP) fields. Under this framework, sensor data is transformed directly into a potential field defined in an image plane where an Image-Based Visual Servoing (IBVS) routine computes control commands to guide the agent toward its goal.

The use of image space has advantages when dealing with the three problems cited above. First, the need for 3D maps is obviated because the agent is no longer reasoning in three dimensions¹. Second, as discussed in [7], image space allows information about arbitrary numbers of agents to be naturally collapsed into single, fixed-size representation. Third, the fact that the representation is fixed-size can simplify the process of developing deterministic control algorithms and defining verification procedures for them.

¹While this prevents reasoning about range or relative velocity explicitly, the agent can reason instead about the ratio of the two, known as *time-to-contact*, which can be extracted directly from a sequence of monocular images [8]



Fig. 1: A dynamically constrained mobile robot agent without prior knowledge of the environment successfully navigates a scene and avoids collision while equipped with only a monocular camera.

This paper expands on [7] by placing the ISP fields in a unitless potential space, rather than directly in the time-to-contact space. This change allows the use of simple composition operations to combine data from arbitrary sources. To ensure that composition does not destroy safety-critical information, two special types of potential transforms are defined. One type specifies soft constraints (such as comfort and goal direction) and the other hard constraints (such as physical and collision), and the subset of ISP space representing hard constraints is shown in [9] to have closure under composition. This property allows, for example, machine learning output to be integrated into safety critical applications, such as automated driving, where it is often impractical for machine learning alone to guarantee hard constraint preservation [10].

The control architecture presented in this paper is built under the Selective Determinism framework presented in [11], which enables robust navigation in dynamic environments through a decomposition of the navigation problem into complementary problems of collision avoidance and guidance. An IBVS subsumption control architecture is chosen to implement selective determinism because it can naturally utilize the ISP fields to generate control commands.

The paper is organized as follows: §II gives a brief background of the techniques used in this paper and of vision-based navigation. §III defines the basic problem and terms that are important, or that have a specific meaning in this work. §IV defines ISP fields and the potential functions used to populate them. §V defines the control framework, followed by §VI, which presents both simulation and real world demonstrations of the concepts presented in this paper. Finally, §VII summarizes and concludes the paper.

II. BACKGROUND

The ISP representation is based on *potential fields* [12]. These fields represent attractive and repulsive forces as scalar fields over a robot’s environment that, at any point, define a force that can be interpreted as a control command. Potential fields have a long history in robotics control. Among recent literature, [13] resembles the approach presented in this paper, but that work places potentials in a Euclidean space, which this approach explicitly avoids.

ISP fields are intended for use in a *subsumption architecture* [14]. In these architectures multiple types of information about a system can be composed together while hard constraint information is *guaranteed* to be preserved. The guarantee in this work is made through a closure property that has been proven to hold under a restricted input space and with specially constructed potential transform functions [9]. In this work, ISP fields are composed in the subsumption process into a single field, where control is accomplished through *visual servoing* techniques [15].

In visual servoing, control is accomplished using feedback computed using image space. This type of control is popular for mobile navigation in aerial agents [16] and in special-purpose uses for marine and land vehicles [17], [18] and manipulation [19]. It, along with other vision-based methods, were also popular for general ground navigation early on in the field of autonomous vehicles [20]. They later fell out of favor in part because of shortcomings in algorithms and hardware. But improvements on both fronts have sparked renewed interest in vision-based methods [21].

The control framework presented in this paper is closely related to Steer Angle Field (SAF) [22] and Vector Field Histogram (VFH) [23] approaches to mobile navigation. Like those, this framework utilizes a decomposition along local collision avoidance and global guidance by choosing a goal-directed control from a set of safe controls computed by the local collision avoidance controller. But this approach differs in two important ways: first, it operates in a potential space rather than measurement space, which can simplify the process of fusing disparate types of information; and second, it performs collision avoidance with respect to time-to-contact rather than distance, which can be a more natural way to handle both moving and stationary obstacles.

III. PROBLEM STATEMENT & DEFINITIONS

The problem this paper attempts to address can be generally stated as follows:

Problem 1. For a mobile agent with multiple sensor modalities, reasonable recognition and tracking algorithms, and limited compute and storage resources, define a control architecture that consumes a fixed amount of resources and enables the agent to navigate towards a goal while remaining collision free.

To aid in presenting the solution to Problem 1, the following definitions are presented:

Definition 1. An *affinely extended potential field* is a potential field with a potential function that ranges over the affinely extended reals $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$. A *positive* (or *negative*) affinely extended potential field is defined over $\overline{\mathbb{R}}$ but contains only positive (or only negative) infinite values.

Definition 2. An *image space potential function* is a mapping of an image pixel value $I(x, y)$ a potential value:

$$I(x, y) \mapsto \overline{\mathbb{R}}$$

Definition 3. *Time-to-contact* (τ), is the predicted duration of time remaining before an object observed by a sensor will come into contact with the reference frame of the sensor.

Definition 4. A *hard constraint* is one that an agent is never allowed to violate, such as geometric or collision constraints. A *soft constraint* is one that an agent is biased not to violate, such as comfort criteria, or learned constraints.

IV. THE IMAGE SPACE POTENTIAL FIELD

Image Space Potential (ISP) fields are affinely extended potential fields that are modeled after image planes. As with image planes, these potential fields can be discretized, and regions of interest (ROIs) can be defined for them. An ISP field is constructed by taking in an “image” of measurement values, which for hard constraints are τ values and for soft constraints are cost or reward values, and then applying the appropriate potential transform to each pixel.

For the ISP field representation to be used in subsumption, it must preserve hard constraint information throughout summation operations. Summation of ISP fields satisfies this requirement provided the following conditions are met:

- 1) All ISP fields involved have like affine extensions.
- 2) ISP fields may only be multiplied by scalars in $(0, +\infty)$.
- 3) ISP fields may only be elementwise multiplied by scalar fields where all values are in $(0, +\infty)$.

These properties guarantee closure during summation for the subsets of hard constraints values in summed ISP fields. For more detailed information and proofs, see [9]. Thus, a control architecture using ISP fields can implement subsumption through addition.

A. The Potential Transforms

Potential transforms encode the difference between hard and soft constraints in terms of the limiting value of the field as the agent approaches some state that would cause constraint violation: the limiting value of the field over measurements where hard constraint violation would occur must be infinite, such that no reward can overwhelm the cost, and the limiting value of the field over measurements where soft constraint violation would occur must be finite, such that a reward must exceed some value before violation occurs.

Many functions can be chosen to satisfy these requirements and be used with this control framework. For details about the specific transforms used in this paper, see [9].

V. VISUAL SERVOING-BASED SUBSUMPTION CONTROL

This section outlines the components and architecture of the control framework presented in this paper.

A. Selective Determinism

Selective Determinism (SD) provides the overall framework within which the control architecture is defined. Under this framework, the navigation problem is interpreted as a guided collision avoidance problem, in which a global guidance controller subsumes a local collision avoidance controller. This is a general framework and is chosen because it has desirable properties when used in multi-agent systems. Specifically, it can guarantee non-collision during navigation provided that each agent in the system:

- 1) Will always choose to avoid collision
- 2) Can estimate the dynamic capabilities of other agents
- 3) Can compute stopping regions for other agents

The stopping region concept of Item 3 is covered in depth in [24]. Intuitively, it is the minimum space within which an agent can come to a stop. The architecture is then built on the following local collision avoidance and global guidance problem definitions:

Problem 2. Local collision avoidance: Assume an agent A navigating a workspace \mathcal{W} receives some observation input O_t of \mathcal{W} at time t . Let \mathcal{A} be the set of objects and agents that does not include A . For metric μ , threshold $\varepsilon > 0$, and history of observations O_i, \dots, O_t , what control set \mathcal{U} is available to A such that $\min \mu > \varepsilon$ for all $u \in \mathcal{U}$ over a time horizon H ?

In this work the metric μ between any two points is taken as the difference between τ and the time needed for the two points to reach zero relative velocity. In many systems, it may be reasonable to assume that so long as $\mu > 0$, the stopping regions for the points are in some way disjoint, or, in other words, that the points can feasibly reach zero relative velocity prior to colliding. This is essentially the time headway rule often used to guide human driving behavior [25].

The global guidance controller problem is then defined as:

Problem 3. Global guidance: For desired goal-directed control u^d , control space metric μ_c , and given a feasible control set \mathcal{U} , choose a control u^* such that:

$$u^* = \arg \min_{u \in \mathcal{U}} \mu_c(u^d, u)$$

Problem 3 subsumes Problem 2 by virtue of the fact that it optimizes over the collision avoidance control set \mathcal{U} .

B. Control Architecture

This section provides a high-level summary of the control architecture. The architecture addresses Problem 2 & Problem 3 independently by computing hard constraint ISP fields that contribute to collision avoidance, and taking in soft constraint ISP fields that contribute to guidance. These fields are added together and a control is computed from the resulting field. This specific control routine is simple and intended

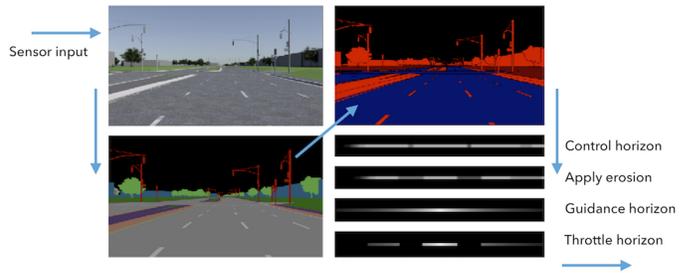


Fig. 2: Camera images (top left) are segmented (bottom left), projected into a potential space (top right), then iteratively transformed into control space (bottom right)

only as a proof of concept; future work will examine more sophisticated controller definitions.

Controls are computed from the final ISP field by iteratively transforming the ISP field into control space through the use of *horizons*, which are scalar fields represented as 1D row vectors. First, a *control horizon* is a $1 \times m$ row vector computed from an $n \times m$ ISP field. The columns of the ISP field are reduced to the control horizon by treating each column in the image as a 1D vector and projecting the entire vector to a scalar; second, an erosion is applied to the control horizon to buffer areas of low potential value; third, a *guidance horizon* is computed that maps yaw values to potential values by interpreting the column index offset from center as yaw; finally, a *throttle horizon* is computed by projecting the previously computed horizons into a $[-1, 1]$ throttle space. The throttle horizon now represents the set of collision avoidance controls \mathcal{U} , where the value at each column index represents the maximum throttle available at the yaw corresponding to that index. The control from this set nearest the desired control (according to a desired metric) is output to the actuators. The process is visualized in Figure 2.

VI. DEMONSTRATIONS

Two demonstrations are run. The first shows how soft constraint values guide navigation. The second demonstration shows the control architecture on a mobile robot that successfully navigates through obstacles.

A. Soft constraint demonstration

This demonstration is performed with publicly available data sets [26] and shows how changes in constraint parameterizations affect the output control space. Figure 3 shows the effect that modifying the user-defined soft constraint values has on the set of controls. The soft constraint values are color-coded in the figure across the top, and the control space output is shown varying with colors along the bottom.

B. Collision avoidance navigation

This demonstration is performed with an open source mobile robot platform [27], and the data recorded during the run has been made publicly available [28]. The experiments demonstrate that control stability can be achieved when the architecture is applied to a real system.

In second demonstration, perception input comes from the fiducial tracking system `ar_track_alvar` [29]. The sequence of detections in image space can be used to calculate estimates of τ for each detection. In [8] it was shown that the size s of a detection in the image plane and its time derivative \dot{s} are sufficient to compute τ . In this work, s is provided directly by the tracking system, and \dot{s} is estimated using simple backward finite differencing over the history of detections. At a time t , for a time step Δt , let $s(t)$ be the scale measurement at time t and $s(t - \Delta t)$ be the previous measurement. It is straightforward to show that τ is:

$$\tau = \frac{s}{\dot{s}} - \Delta t \quad (1)$$

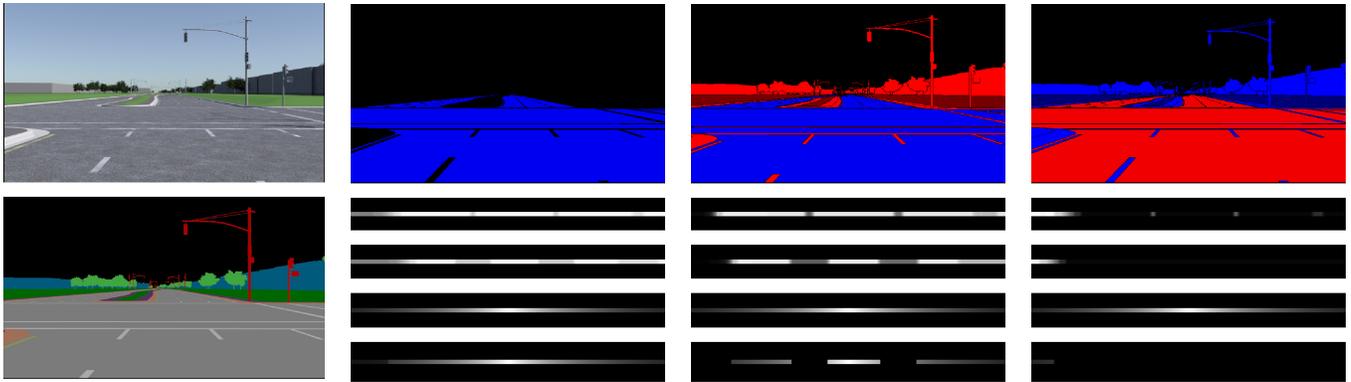
Figure 4 describes the experiment and shows the mobile test platform as it navigates the slalom (link to video in caption).

VII. CONCLUSIONS

This paper presented Image Space Potential (ISP) fields and a visual servoing-based subsumption control architecture for mobile navigation. ISP fields can obviate the need for geometric maps, they are constant space complexity with respect to the image, and they are amenable to verification. While similar to SAF and VFH approaches, this approach deals naturally with both moving and stationary obstacles, and, under reasonable assumptions, the formulation of the control architecture can also ensure collision-free navigation in multi-agent systems. An implementation of the framework described in this paper is publicly available [30] under the MIT open source license [31].

REFERENCES

- [1] "Whoever Owns the Maps Owns the Future of Self-Driving Cars," <https://is.gd/uxagoh>, accessed: 2018-06-23. 1
- [2] "Nobody Wants to Let Google Win the War for Maps All Over Again," <https://is.gd/otesud>, accessed: 2018-06-23. 1
- [3] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic mdp-behavior planning for cars," in *14th International IEEE Conference on Intelligent Transportation Systems, ITSC 2011, Washington, DC, USA, October 5-7, 2011*, 2011, pp. 1537–1542. [Online]. Available: <https://doi.org/10.1109/ITSC.2011.6082928> 1
- [4] F. A. Oliehoek and C. Amato, *A Concise Introduction to Decentralized POMDPs*, ser. Springer Briefs in Intelligent Systems. Springer, 2016. 1
- [5] S. Louise, V. David, J. Delcoigne, and C. Aussaguès, "OASIS project: deterministic real-time for safety critical embedded systems," in *Proceedings of the 10th ACM SIGOPS European Workshop, Saint-Emilion, France, July 1, 2002*, 2002, pp. 223–226. [Online]. Available: <http://doi.acm.org/10.1145/1133373.1133419> 1
- [6] "Road vehicles – Functional safety," International Organization for Standardization, Geneva, CH, Standard, Nov. 2011. 1
- [7] J. K. Johnson, "Constant space complexity environment representation for vision-based navigation," in *IROS 2017 9th Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, 2017. 1
- [8] T. Camus, "Real-time optical flow," Ph.D. dissertation, Brown University, Department of Computer Science, 1994. 1, 4
- [9] J. K. Johnson, "Image Space Potential Fields: Constant Size Environment Representation for Vision-based Subsumption Control Architectures," Maeve Automation, Tech. Rep., September 2017. [Online]. Available: <https://arxiv.org/abs/1709.09662> 1, 2
- [10] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *CoRR*, vol. abs/1610.03295, 2016. [Online]. Available: <http://arxiv.org/abs/1610.03295> 1
- [11] J. K. Johnson, "Selective determinism for autonomous navigation in multi-agent systems," Ph.D. dissertation, Indiana University, School of Informatics, Computing, and Engineering, September 2017. <http://hdl.handle.net/2022/21670>. 1
- [12] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings of the 1985 IEEE International Conference on Robotics and Automation, St. Louis, Missouri, USA, March 25-28, 1985*, pp. 500–505. [Online]. Available: <https://doi.org/10.1109/ROBOT.1985.1087247> 2
- [13] M. T. Wolf and J. W. Burdick, "Artificial potential functions for highway driving with collision avoidance," in *2008 IEEE International Conference on Robotics and Automation, ICRA 2008, May 19-23, 2008, Pasadena, California, USA*, 2008, pp. 3731–3736. [Online]. Available: <https://doi.org/10.1109/ROBOT.2008.4543783> 2
- [14] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986. [Online]. Available: <https://doi.org/10.1109/JRA.1986.1087032> 2
- [15] L. E. Weiss, A. C. Sanderson, and C. P. Neuman, "Dynamic visual servo control of robots: An adaptive image-based approach," in *Proceedings of the 1985 IEEE International Conference on Robotics and Automation, St. Louis, Missouri, USA, March 25-28, 1985*, pp. 662–668. [Online]. Available: <https://doi.org/10.1109/ROBOT.1985.1087296> 2
- [16] M. A. O.-M. Pascual Campoy, Ivan F. Mondragon and C. Martinez, *Visual Servoing*. IntechOpen, 2010, ch. Visual Servoing for UAVs. 2
- [17] S. Benhimane and E. Malis, "Homography-based 2d visual tracking and servoing," *I. J. Robotics Res.*, vol. 26, no. 7, pp. 661–676, 2007. [Online]. Available: <https://doi.org/10.1177/0278364907080252> 2
- [18] Y.-H. Kim, S.-W. Lee, H. S. Yang, and D. A. Shell, "Toward autonomous robotic containment booms: visual servoing for robust inter-vehicle docking of surface vehicles," *Intelligent Service Robotics*, vol. 5, no. 1, pp. 1–18, Jan 2012. [Online]. Available: <https://doi.org/10.1007/s11370-011-0100-0> 2
- [19] D. Kragic and H. I. Christensen, "Survey on visual servoing for manipulation," COMPUTATIONAL VISION AND ACTIVE PERCEPTION LABORATORY, Tech. Rep., 2002. 2
- [20] V. Graefe and K.-D. Kuhnert, *Vision-based Autonomous Road Vehicles*. New York, NY: Springer New York, 1992, pp. 1–29. [Online]. Available: https://doi.org/10.1007/978-1-4612-2778-6_1 2
- [21] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. D. Jackel, and U. Muller, "Explaining how a deep neural network trained with end-to-end learning steers a car," *CoRR*, vol. abs/1704.07911, 2017. [Online]. Available: <http://arxiv.org/abs/1704.07911> 2
- [22] R. Bauer, W. Feiten, and G. Lawitzky, "Steer angle fields: An approach to robust manoeuvring in cluttered, unknown environments," *Robotics and Autonomous Systems*, vol. 12, no. 3-4, pp. 209–212, 1994. [Online]. Available: [https://doi.org/10.1016/0921-8890\(94\)90027-2](https://doi.org/10.1016/0921-8890(94)90027-2) 2
- [23] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Trans. Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991. [Online]. Available: <https://doi.org/10.1109/70.88137> 2
- [24] J. K. Johnson, "On the relationship between dynamics and complexity in multi-agent collision avoidance," *Autonomous Robots*, Apr 2018. [Online]. Available: <https://doi.org/10.1007/s10514-018-9743-4> 3
- [25] R. Lamm, B. Psarianos, and T. Maillaender, *Highway Design and Traffic Safety Engineering Handbook*, ser. McGraw-Hill handbooks. McGraw-Hill, 1999. [Online]. Available: <https://books.google.com/books?id=mNBnQgAACAAJ> 3
- [26] K. McNamara, "Parallel Domain Simulation Sequences," <https://maeveautomation.com/data-sets/>, 2017. 3, 5
- [27] W. Roscoe, *Donkey Car: An open source DIY self driving platform for small scale cars*. [Online]. Available: <http://www.donkeycar.com> 3
- [28] J. K. Johnson, "Maeve Experiment Recordings," <https://maeveautomation.com/data-sets/>, 2017. 3
- [29] S. Niekum, "ar_track_alvar: A ROS wrapper for Alvar, an open source AR tag tracking library," https://github.com/ros-perception/ar_track_alvar, 2017. 4
- [30] J. K. Johnson, "Maeve Automation Development Libraries," https://bitbucket.org/maeveautomation/maeve_automation_core, 2017. 4
- [31] "The MIT License," Massachusetts Institute of Technology. [Online]. Available: <https://opensource.org/licenses/MIT> 4



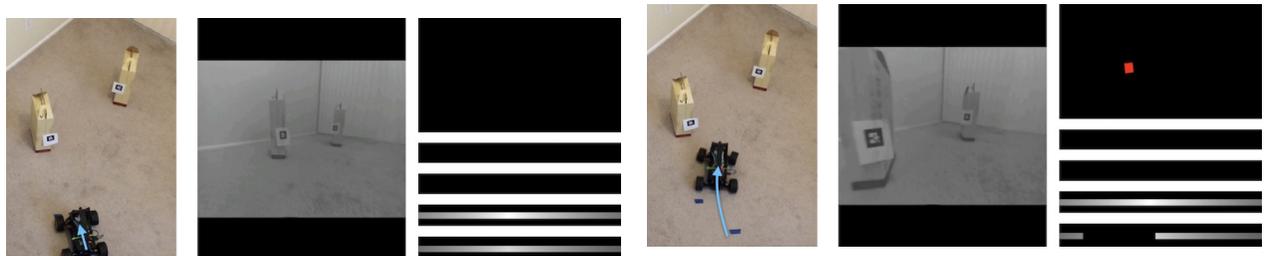
(a) Top: The raw camera image input. This image passes through a perception routine that can produce a segmentation. Bottom: The segmentation output of perception.

(b) Road pixels have positive bias (blue), all others have neutral (black). The final horizon (bottom) essentially mirrors the guidance horizon (second from bottom).

(c) Road pixels have positive bias (blue), while lane markers have negative (red). The final horizon (bottom) strongly avoids forward motion that aligns with lane boundaries.

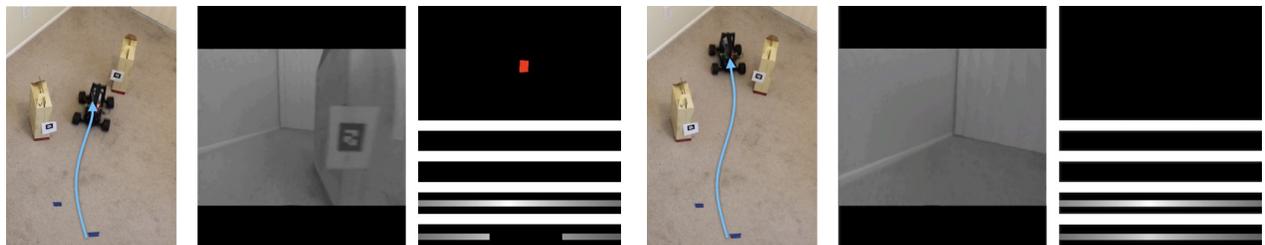
(d) Road pixels have negative bias (red), while lane markers have positive (blue). The final horizon (bottom) strongly avoids virtually all forward motion save for a faint patch at the far left.

Fig. 3: Demonstration 1: The leftmost column shows the input to the control architecture. From there left to right are various soft constraint configurations. From top to bottom in each of those columns are the ISP field visualization and the sequence of horizons described in §V-B and shown in Figure 2. The horizons visualize available throttles for for each yaw angle corresponding to an image column, where positive throttle is lighter and negative throttle is darker. This data set, and others, are publicly available [26]. Best viewed in color.



(a) Initial approach of the agent to the obstacles. The bias brings the vehicle towards the near obstacle. At this point the agent is stationary, so the ISP field (top right) is empty.

(b) Once the agent begins moving forward, the near obstacle induces an increasing negative potential region in the ISP field (top right). As the potential drops, the controls that steer the agent toward the obstacle are made unavailable (bottom right).



(c) As a result of avoiding the first obstacle, the agent approaches the second obstacle, which induces a different portion of the control set to become unavailable (bottom right).

(d) Once the agent moves past the obstacles, the ISP field becomes empty again, and the bias value guides control unperturbed.

Fig. 4: Demonstration 2: A global guidance control always directs full throttle and centered steering. The agent uses ISP fields to move in a way that is biased toward the guidance control while taking evasive action to avoid obstacles. Best viewed in color. Video available at: <https://youtu.be/BaHekhZmkfY>.