

Self-Supervised Online Learning for Safety-Critical Control using Stereo Vision

Ryan K. Cosner*, Ivan D. Jimenez Rodriguez*, Tamas G. Molnar, Wyatt Ubellacker,
Yisong Yue, Aaron D. Ames, and Katherine L. Bouman

Abstract—With the increasing prevalence of complex vision-based sensing methods for use in obstacle identification and state estimation, characterizing environment-dependent measurement errors has become a difficult and essential part of modern robotics. This paper presents a self-supervised learning approach to safety-critical control. In particular, the uncertainty associated with stereo vision is estimated, and adapted online to new visual environments, wherein this estimate is leveraged in a safety-critical controller in a robust fashion. To this end, we propose an algorithm that exploits the structure of stereo-vision to learn an uncertainty estimate without the need for ground-truth data. We then robustify existing Control Barrier Function-based controllers to provide safety in the presence of this uncertainty estimate. We demonstrate the efficacy of our method on a quadrupedal robot in a variety of environments. When not using our method safety is violated. With offline training alone we observe the robot is safe, but overly-conservative. With our online method the quadruped remains safe and conservatism is reduced.

I. INTRODUCTION

Accounting for vision-based uncertainty is particularly important for modern safety-critical robotic applications such as autonomous vehicles, health care, and manufacturing [1]. Such safety-critical systems require controllers that provide robust safety in the presence uncertainty. *Control Barrier Functions (CBFs)* [2], [3] are a popular tool which can be used to guarantee safety through the satisfaction of a Lyapunov-like inequality. CBFs have been used to achieve several safety-critical robotic tasks such as obstacle avoidance [4], multi-agent navigation [5], and safe walking [6]. However, standard CBF theory requires accurate state estimation. This motivates the need for a method that provides safety in the presence of noisy sensor measurements.

Computer vision has become an important tool in robotics for sensing environments and identifying obstacles. It is often an integral component of robotics applications such as simultaneous localization and mapping (SLAM [7]). Despite the utility and ubiquity of computer vision, using vision sensors to achieve robust safety is difficult due to the complex environment-dependent error that they generate. For example, error patterns are highly correlated with the textures and appearance of a scene. Supervised methods

* These authors contributed equally.

This research is supported in part by the National Science Foundation CPS Award #1932091, Dow (#227027AT), BP p.l.c., AeroVironment.

Authors are with the Department of Control and Dynamical Systems, the Department of Mechanical and Civil Engineering, and with the Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125, USA, {rkc cosner, ivan.jimenez, tmolnar, wubellac, yyue, ames, klbouman}@caltech.edu.

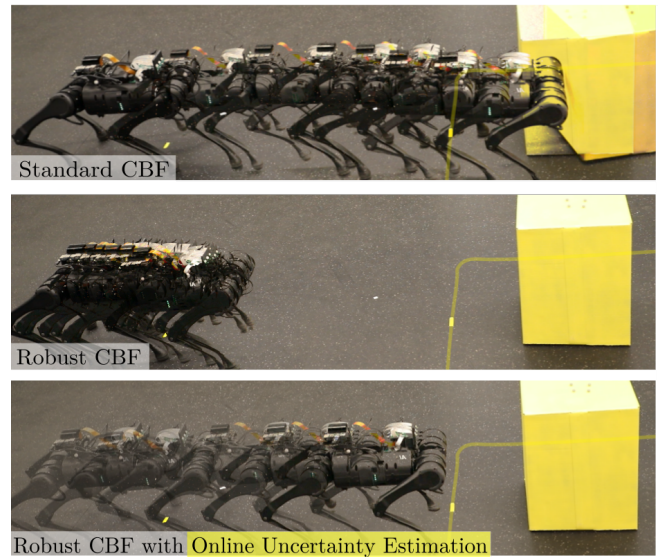


Fig. 1. These space-time images display our quadrupedal robot throughout the course of an experiment. The robot is considered safe if it remains left of the yellow line. The standard control barrier function (CBF) condition fails to keep the robot safe due to errors in stereo vision; the robust CBF condition keeps the robot safe, but is conservative; and our proposed method, Robust CBFs with Online Uncertainty Estimation, keeps the robot safe without remaining overly conservative.

can identify and model error as it affects the CBF [8], [9], [6], [10], [11]; however, supervised approaches require ground-truth training data that may be difficult or impossible to obtain. Additionally, such frameworks often inaccurately estimate errors in regions of the state space that were under-sampled in the training set; training on indoor environments or synthetic data with easy-to-access ground-truth data often does not translate well to outdoor environments.

In this paper, we focus on two main challenges related to automatic vision-based safety critical control: (i) accounting for the effect of perception uncertainty on the controller, and (ii) estimating that uncertainty while operating in novel environments. We tackle these issues in the context of stereo vision-based obstacle avoidance on a quadrupedal robot.

Namely, our goal is to avoid obstacles seen in stereo cameras mounted on the robot, while accounting for the uncertainty of vision-based measurements. To do that we use a multibaseline stereo vision system on the robot to record stereo images. We then determine the position of the objects associated with each image pixel, and subsequently infer the uncertainty of these positions through a self-supervised error estimation algorithm that frames the problem as online learning [12]. Online learning frameworks have shown

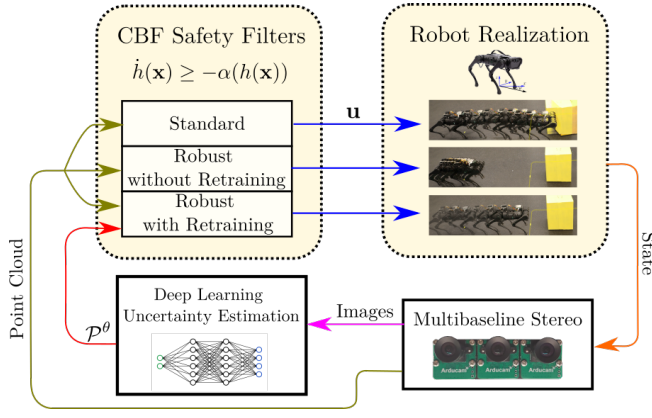


Fig. 2. The overarching structure of our approach. It begins on the bottom right by capturing three time-synchronized images that are then fed into an uncertainty estimation pipeline and also used to generate a 3D point cloud. There are three possible CBF filters that result in the three robot realizations shown. From top to bottom, the standard filter only takes into account the noisy point-cloud in avoiding obstacles. The robust CBF safety filters use the estimate of the uncertainty \mathcal{P} to compensate for noise in the point cloud. Finally, the “Robust with Retraining” filter refines the model of uncertainty to the current environment in real-time.

success in a variety of robotic applications [13], [14], [15]. Finally, we use the position and uncertainty estimates for safety critical control, wherein we achieve robust safety via CBFs. A visualization of this method can be found in Fig. 2.

The contributions of this work are three-fold. First, we present and evaluate an online, self-supervised method for characterizing the uncertainty of disparity errors generated by stereo vision algorithms in novel environments (Section II). Second, we develop a robustified CBF-based control method which utilizes this error estimate for obstacle avoidance (Section III). And third, we demonstrate the proposed methods of error estimation and obstacle avoidance on a quadrupedal robot operating in real time (Section IV).

II. STEREO VISION UNCERTAINTY QUANTIFICATION

We begin by revisiting stereo vision-based depth estimation. We then propose an approach for learning the uncertainty of a black box stereo-matching algorithm. The proposed self-supervised learning approach can be trained online and takes advantage of geometric structure in stereo disparity maps so as not to require ground truth data.

A. Background in Stereo Vision

Stereo vision is a popular tool for determining depth from images. These methods compute a *disparity*: the shift observed in an object’s projection onto two camera planes. Using a geometric understanding of the camera setup, pixel-based disparity maps can be converted to depth maps. Errors in the final depth-map result from a combination of pixel-mismatch in disparity estimation and error in the camera parameters used to convert from disparity to depth. The errors in the intrinsic and extrinsic parameters of the camera are usually small and their effect on the resulting depth distribution is easy to compute. On the other hand, pixel matching errors are much larger and are the result of a

much more complicated stereo matching procedure whose effect on the resulting disparity is difficult to quantify and environment-dependent.

For standard stereo vision we adopt the model of [16] for two cameras (left and right) and assume that they are perfectly rectified, vertically aligned and evenly spaced with known distance $b \in \mathbb{R}_{>0}$ between each camera. Pixel coordinates within an image are given by the tuple $p \triangleq (u, v) \in K$, where $K \triangleq \{0, \dots, W\} \times \{0, \dots, H\}$ for image width $W \in \mathbb{N}_{>0}$ and image height $H \in \mathbb{N}_{>0}$.

Stereo algorithms such as Block Matching, Semi-Global Block Matching, and Efficient Large-Scale Stereo [17] compute disparities by determining the discrete pixel distance between matching regions of two images. Since the disparity represents a shift between pixels of two images, the measured disparity \hat{d} must be a finite integer value. Assuming that the true disparity d is a finite integer implies that the error $e \triangleq \hat{d} - d$ must also be a finite integer. Prior work has been done to interpolate disparities for non-integer subpixel accuracy [18]; however, we restrict our attention to integer disparity values to highlight the error in pixel-matching.

B. Self-supervised Error Estimation

To learn the error in disparity, we introduce a three-camera multibaseline stereo system which produces multiple disparity maps that are related through simple functions; deviations from the ideal relationship indicate error in the estimated disparities. By analyzing the correlation of image appearance with these errors, a function that estimates disparity error from appearance is learned and used to specify state error-bounds in real-time for use in a robustified CBF.

We introduce a three-element camera system, whose central camera is assumed to be perfectly rectified and vertically aligned with the other two cameras as shown in Fig. 2. This third camera is placed between the left and right cameras such that it has a baseline of $b/2$ with both. The three cameras produce a time-synchronized grayscale image triple (I_1, I_2, I_3) where $I_i \in \mathbb{N}^{W \times H}$ for $i \in 1, 2, 3$ and 1, 2, 3 correspond with left, center, and right, respectively. The disparity between any image pair (I_i, I_j) for $i < j$ is obtained using the stereo-vision algorithm $\mathcal{D} : \mathbb{N}^{W \times H} \times \mathbb{N}^{W \times H} \rightarrow \Gamma^{W \times H}$, so that $\hat{d}_{i,j} = \mathcal{D}(I_i, I_j)$. Here, $\Gamma \subset \mathbb{N}_{\geq 0}$ is the set of possible disparity values.

Given the measurement $\hat{d}_{i,j}$, the error appears as $\hat{d}_{i,j} = d_{i,j} + e_{i,j}$ with error distribution $e_{i,j} \sim \mathcal{P}(I_i, I_j)$ and ground truth disparity $d_{i,j} \in \Gamma^{W \times H}$. We model this error as a discrete random variable with probability $\mathcal{P}(I_i, I_j)$ on $\Gamma^{W \times H}$. This model of disparity errors contrasts sharply with other common error models, such as punctual observation, uniform observation, and Gaussian observation [16], in that it accounts for the discrete nature of stereo-pixel matching algorithms. If ground-truth knowledge of $d_{i,j}$ is obtainable, then supervised learning methods can be implemented to directly estimate this error term. However, it is often the case that ground-truth knowledge is unavailable; particularly when a domain transfer must occur during operation. Thus

we seek a general method to estimate $e_{i,j}$ for any black-box disparity algorithm without the need for ground-truth data.

We leverage the known geometric relationships between the three cameras to learn a mapping between image appearance and disparity error distribution that can adapt during operation in new environments. Given a multibaseline stereo system, if one ignores occlusions, it is possible to completely reconstruct each disparity map from the other two maps. The relationship to reconstruct $\hat{d}_{1,3}$ from $\hat{d}_{1,2}$ and $\hat{d}_{2,3}$ is shown in Algorithm 1; we denote this reconstruction as $\bar{d}_{1,3} \triangleq \hat{d}_{1,2} \oplus \hat{d}_{2,3}$.

Algorithm 1 Disparity Reconstruction: $\bar{d}_{1,3} = \hat{d}_{1,2} \oplus \hat{d}_{2,3}$

```

1:  $\bar{d}_{1,3} \leftarrow \mathbf{0}_{H \times W}$ 
2: for  $v \in [1, \dots, H]$  do
3:   for  $u \in [1, \dots, W]$  do
4:      $\hat{u} \leftarrow n + \hat{d}_{1,2}(u, v)$ 
5:      $\bar{d}_{1,3}(u, v) \leftarrow \hat{d}_{1,2}(u, v) + \hat{d}_{2,3}(u, \hat{u})$ 
6:   end for
7: end for
```

We use the reconstructed disparity $\bar{d}_{1,3}$ to learn the parameters θ of a function \mathcal{P}^θ that approximates error distribution \mathcal{P} (refer to Algorithm 2). Since this method does not require ground truth information, Algorithm 2 can be run online during operation to adapt \mathcal{P}^θ to new visual environments. Recall that the disparity error, $e_{1,3}$ is discrete in nature. Therefore, the pixel-wise reconstruction error $re(p) \triangleq \|\hat{d}_{1,3}^p - \bar{d}_{1,3}^p\|_1$ will also be discrete. For this reason, optimizing the loss L reduces to a pixel-wise classification problem similar to image segmentation. Thus, as is done in image segmentation, we use pixel-wise cross entropy as the loss function L . This method is shown in Algorithm 2. In Line 8, for each pixel p

Algorithm 2 Self-Supervised Stereo Error Estimation Adaptation

```

1:  $L \leftarrow 0$ 
2: while robot is running do
3:    $(I_1, I_2, I_3) \leftarrow \text{Capture Current Frame}$ 
4:    $\hat{d}_{1,2} \leftarrow \mathcal{D}(I_1, I_2)$ 
5:    $\hat{d}_{2,3} \leftarrow \mathcal{D}(I_2, I_3)$ 
6:    $\hat{d}_{1,3} \leftarrow \mathcal{D}(I_1, I_3)$ 
7:    $\bar{d}_{1,3} \leftarrow \hat{d}_{1,2} \oplus \hat{d}_{2,3}$ 
8:    $re(p) \leftarrow \|\hat{d}_{1,3}^p - \bar{d}_{1,3}^p\|_1$ 
9:    $L \leftarrow -\frac{1}{H \times W} \sum_p \mathbb{E}_{\mathbf{1}(re(p))} [\log \mathcal{P}^\theta(I_i, I_k)]$ 
10:   $\theta_{t+1} \leftarrow \theta_t - \eta \frac{\partial L}{\partial \theta}$ 
11: end while
```

of the disparity $\hat{d}_{1,3}$ the corresponding reconstruction error is computed. The loss function in Line 9 is then equivalent to the expected negative log likelihood of each pixel under the proposed model \mathcal{P}^θ . An example visualization of lines 3–8 can be found in Fig. 3. Although this algorithm focuses on the reconstructed disparity $\bar{d}_{1,3}$, it can be easily extended to similar reconstructions of $d_{1,2}$ and $d_{2,3}$.

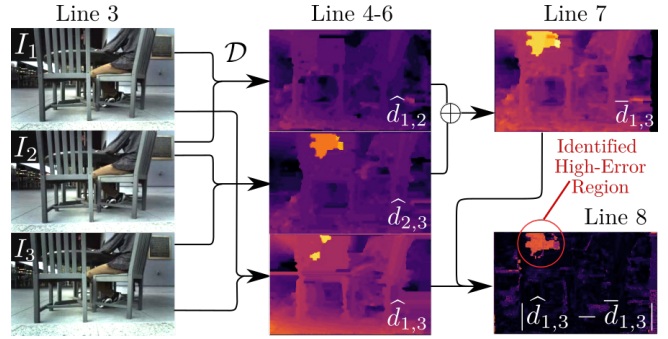


Fig. 3. Lines 3-8 of Algorithm 2 illustrated from left to right. Starting from three time-synchronized images three pairwise disparities are computed as shown in the middle column. Two of these disparities are used to build a reconstruction of the third disparity shown in the top right which can then be used to estimate the pixel-wise error of the stereo algorithm shown in the bottom right image. These steps of the algorithm correctly identify that the back of the closest chair is a high-error region without using ground truth information. This information is used to learn a correspondence between visual features and error distributions.

Supervised methods have been used in the past to estimate uncertainty in robotic applications by computing the covariance of state estimates [19]. Our approach differs in that we do not require ground truth and we take advantage of the discrete structure of images to learn a discrete, rather than a Gaussian, distribution, which is better suited to the disparity measurements of stereo vision.

III. SAFE VISION-BASED CONTROL

In this section we review Control Barrier Functions (CBFs) [20] as a tool for guaranteeing the safety of dynamical systems. We then propose CBFs that rely on the position of pixels provided by stereoscopic sensing. Finally, we incorporate the proposed self-supervised error estimates of Section II to enforce robust safety.

A. Control Barrier Functions

First we give a brief introduction to CBFs which follows our description in [21], where additional technical details can be found. In this work we consider the safety of robotic systems with control affine dynamics

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \quad \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{u} \in \mathbb{R}^m, \quad (1)$$

where \mathbf{x} is the state of the system, \mathbf{u} is the input, $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the drift dynamics, and $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ is the input matrix. We assume that \mathbf{f} and \mathbf{g} are locally Lipschitz continuous. Given a locally Lipschitz continuous state-feedback controller $\mathbf{k} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, the closed-loop dynamics are governed by:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{k}(\mathbf{x}). \quad (2)$$

For any initial condition $\mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^n$ there exists a unique solution $\mathbf{x}(t)$ to (2), which we assume to exist $\forall t \in [0, \infty)$.

The notion of safety is formalized by defining a *safe set* $\mathcal{C} \subset \mathbb{R}^n$ in the state space that the system must remain within. In particular, consider the set \mathcal{C} as the 0-superlevel set of a continuously differentiable function $h : \mathbb{R}^n \rightarrow \mathbb{R}$:

$$\mathcal{C} \triangleq \{\mathbf{x} \in \mathbb{R}^n \mid h(\mathbf{x}) \geq 0\}, \quad (3)$$

where $h(\mathbf{x}) = 0 \implies \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}) \neq 0$ and \mathcal{C} is non-empty and has no isolated points. Safety is defined as the *forward invariance* of \mathcal{C} , i.e., if $\mathbf{x}_0 \in \mathcal{C}$, then $\mathbf{x}(t) \in \mathcal{C}$ for all $t \geq 0$.

To synthesize controllers that ensure safety, we use Control Barrier Functions (CBFs) [2] defined as follows:

Definition 1 (Control Barrier Function (CBF)). Let $\mathcal{C} \subset \mathbb{R}^n$ be a safe set given by (3). The function h is a *Control Barrier Function* (CBF) for (1) on \mathcal{C} if there exists $\gamma \in \mathcal{K}_{\infty,e}$ ¹ such that for all $\mathbf{x} \in \mathcal{C}$:

$$\sup_{\mathbf{u} \in \mathbb{R}^m} \dot{h}(\mathbf{x}, \mathbf{u}) \triangleq \underbrace{\frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}) \mathbf{f}(\mathbf{x})}_{L_{\mathbf{f}}h(\mathbf{x})} + \underbrace{\frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}) \mathbf{g}(\mathbf{x}) \mathbf{u}}_{L_{\mathbf{g}}h(\mathbf{x})} \geq -\gamma(h(\mathbf{x})), \quad (4)$$

where $L_{\mathbf{f}}h : \mathbb{R}^n \rightarrow \mathbb{R}$ and $L_{\mathbf{g}}h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ are the Lie derivatives of h with respect to \mathbf{f} and \mathbf{g} respectively.

A main result in [20], [22] relates CBFs to the safety of the closed-loop system (2) with respect to \mathcal{C} :

Theorem 1. *Given a safe set $\mathcal{C} \subset \mathbb{R}^n$, if h is a CBF for (1) on \mathcal{C} , then any locally Lipschitz continuous controller $\mathbf{k} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ satisfying*

$$L_{\mathbf{f}}h(\mathbf{x}) + L_{\mathbf{g}}h(\mathbf{x})\mathbf{k}(\mathbf{x}) \geq -\gamma(h(\mathbf{x})) \quad (5)$$

for all $\mathbf{x} \in \mathcal{C}$, renders the system (2) safe w.r.t. \mathcal{C} .

Given a nominal (but not necessarily safe) locally Lipschitz continuous controller $\mathbf{k}_d : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and a CBF h , the CBF-Quadratic Program (CBF-QP) [2] is a controller that guarantees the system's safety:

$$\begin{aligned} \mathbf{k}(\mathbf{x}) = \underset{\mathbf{u} \in \mathbb{R}^m}{\operatorname{argmin}} \quad & \frac{1}{2} \|\mathbf{u} - \mathbf{k}_d(\mathbf{x})\|_2^2 \\ \text{s.t.} \quad & L_{\mathbf{f}}h(\mathbf{x}) + L_{\mathbf{g}}h(\mathbf{x})\mathbf{u} \geq -\gamma(h(\mathbf{x})). \end{aligned} \quad (\text{CBF-QP})$$

B. Control Barrier Functions for Safe Vision-Based Control

Next we apply CBFs to achieve safe obstacle avoidance for robotic systems based on stereo vision. First we construct CBFs for safe vision-based control. Let $\rho_p \in \mathbb{R}^3$ represent the true three-dimensional position of the portion of the scene which generated pixel p . Using this, we can define a CBF $h : \mathbb{R}^n \times \mathbb{R}^3 \rightarrow \mathbb{R}$ that relies on both the state \mathbf{x} and three dimensional pixel position ρ_p . The pixel position is a geometric function of the true disparity, $\rho_p = T(\mathbf{x}, r(p, d_{1,3}^p))$ where $r : \mathbb{N}^2 \times \mathbb{N}$ is the stereo reprojection function and $T : \mathbb{R}^n \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is the transformation mapping from the robot's state and relative pixel position to pixel position.

In order to relate the output of the stereoscopic sensor with safety, we make the following assumptions:

Assumption 1. The environment is static, so the time derivative of the pixelized environment is zero: $\frac{d\rho_p}{dt} = 0$ for all $p \in K$.

Assumption 2. Ensuring safety with respect to the true three dimensional pixel locations is sufficient to ensure safety with respect to the environment. That is, the safe set for the system is given by:

$$\mathcal{C}_K = \{\mathbf{x} \in \mathbb{R}^n \mid h(\mathbf{x}, \rho_p) \geq 0, \forall p \in K\} \quad (6)$$

¹ $\mathcal{K}_{\infty,e}$ denotes the set of extended class- \mathcal{K} infinity functions, wherein $\gamma \in \mathcal{K}_{\infty,e}$ satisfies $\gamma : \mathbb{R} \rightarrow \mathbb{R}$ is strictly increasing, $\gamma(0) = 0$, and $\lim_{r \rightarrow \infty} \gamma(r) = \infty$, $\lim_{r \rightarrow -\infty} \gamma(r) = -\infty$.

where $h : \mathbb{R}^n \times \mathbb{R}^3 \rightarrow \mathbb{R}$ is the CBF for pixel p .

Although it is not outlined in this work, Assumption 1 can be relaxed to include moving environments by calculating \dot{h} accordingly and estimating the motion of obstacles in the environment. Assumption 2 simplifies the surrounding environment from infinite- to finite-dimensional by assuming that the environment is smooth between a sufficiently dense coverage of pixels. It also implies that the system only has to stay safe with respect to objects that can be seen in the cameras' field of view.²

Given Assumptions 1 and 2 and Theorem 1, synthesizing the control input \mathbf{u} such that

$$L_{\mathbf{f}}h(\mathbf{x}, \rho_p) + L_{\mathbf{g}}h(\mathbf{x}, \rho_p)\mathbf{u} \geq -\gamma(h(\mathbf{x}, \rho_p)), \quad (7)$$

$\forall p \in K$, is sufficient to guarantee safety. Considering each pixel $p \in K$, however, may be computationally intractable, therefore we seek a condition with fewer required constraints.

To combine the constraints, we apply Boolean composition to each CBF h to produce a single nonsmooth CBF h_{ns} ,

$$h_{\text{ns}}(\mathbf{x}) \triangleq \min_{p \in K} h(\mathbf{x}, \rho_p), \quad (8)$$

and simply enforce the CBF constraint associated with the pixels whose CBFs have the smallest value [23]. In particular, to achieve safety it is sufficient to enforce only the constraints whose indices appear in the locally-encapsulating index set:

$$\Lambda = \{p \in K : h(\mathbf{x}, \rho_p) \leq h_{\text{ns}}(\mathbf{x}) + \delta\}, \quad (9)$$

for some $\delta > 0$, as stated formally below.

Theorem 2 ([23], Prop III.6). *Let $h : \mathbb{R}^n \times \mathbb{R}^3 \rightarrow \mathbb{R}$ be a locally Lipschitz function and h_{ns} be as in (8). If there exists a locally Lipschitz extended class \mathcal{K} function γ and a measurable and locally bounded controller $\mathbf{k} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ that satisfies:*

$$\min_{p \in \Lambda} \{L_{\mathbf{f}}h(\mathbf{x}, \rho_p) + L_{\mathbf{g}}h(\mathbf{x}, \rho_p)\mathbf{k}(\mathbf{x})\} \geq -\gamma(h_{\text{ns}}(\mathbf{x})). \quad (10)$$

Then h_{ns} is a valid nonsmooth CBF and the closed loop dynamics (2) with controller \mathbf{k} are safe with respect to \mathcal{C}_K .

This theorem indicates that enforcing the CBF condition only for the “least safe” pixel is sufficient to guarantee the safety of the system.

C. Robustness to Uncertainty

Error in the disparity propagates to the controller in the form of the measured 3D pixel position $\hat{\rho}_p$. The measured value $\hat{\rho}_p$ lies in a neighborhood \mathcal{E}_p of the true value ρ_p , which is characterized by the error distribution $\mathcal{P}(I_i, I_j)$. We assume that the distribution $\mathcal{P}(I_i, I_j)$ is symmetric about the measured value and define the pixel-wise uncertainty set:

$$\mathcal{E}_p \triangleq \left\{ \rho \in \mathbb{R}^3 \mid \begin{array}{l} \rho = T(\mathbf{x}, r(p, \xi)), \quad \xi \in \Gamma \\ \mathcal{P}^\theta(e_{1,3}(p) < |\xi - \hat{d}(p)|; I_1, I_3) \geq \sigma \end{array} \right\} \quad (11)$$

²The field of view aspect of Assumption 2 can be overcome by tracking features that leave the frame as done in Simultaneous Localization and Mapping (SLAM) algorithms [7].

where $\sigma > 0$ is a parameter defining the desired uncertainty robustness.

To achieve safety, one must determine which pixels are safety-critical given \mathcal{E}_p and then enforce robust safety with respect to those pixels. The safety-critical pixels can be determined by expanding the index set Λ using the uncertainty:

$$\Lambda \subseteq \left\{ p \in K \mid h(\mathbf{x}, \rho_p) \leq \max_{\rho_p \in \mathcal{E}_p} \min_{p \in K} h(\mathbf{x}, \rho_p) + \delta \right\}. \quad (12)$$

This can further be expanded to an easily calculable index set $\hat{\Lambda} \supseteq \Lambda$ by minimizing the left-hand-side of the inequality condition and using the max-min inequality [24]:

$$\hat{\Lambda} = \left\{ p \in K \mid \min_{\rho_p \in \mathcal{E}_p} h(\mathbf{x}, \rho_p) \leq \min_{p \in K} \max_{\rho_p \in \mathcal{E}_p} h(\mathbf{x}, \rho_p) + \delta \right\}. \quad (13)$$

This expanded index set $\hat{\Lambda}$ accounts for uncertainty and indicates which pixels are safety-critical and which constraints must be enforced to achieve safety given the pixel-wise uncertainty sets \mathcal{E}_p .

Measurement-Robust Control Barrier Functions (MR-CBFs) as outlined in [25] are a general method for accounting for state uncertainty in CBFs. We can use this method for each pixel $p \in \hat{\Lambda}$ to ensure that the safety constraint is satisfied despite the uncertainty. The resulting constraint is:

$$\begin{aligned} & L_f h(\mathbf{x}, \hat{\rho}_p) + L_g h(\mathbf{x}, \hat{\rho}_p) \mathbf{u} \\ & - (\mathcal{L}_{L_f h} + \mathcal{L}_{\gamma \circ h_{ns}} + \mathcal{L}_{L_g h} \|\mathbf{u}\|_2) \epsilon_p \\ & \geq -\gamma(h_{ns}(\mathbf{x})), \quad \forall p \in \hat{\Lambda} \end{aligned} \quad (14)$$

where \mathcal{L} is the Lipschitz constant of the subscript and

$$\epsilon_p \geq \max_{\rho_p \in \mathcal{E}_p} \|\rho_p - \hat{\rho}_p\|_2 \quad (15)$$

is a bound on the uncertainty. Since $\Lambda \subseteq \hat{\Lambda}$ and the MR-CBF condition implies the CBF condition (5), satisfying (14) also satisfies (10) providing safety of the system if $\sigma = 1$ and $\mathcal{P}^\theta = \mathcal{P}$.

IV. APPLICATION: OBSTACLE AVOIDANCE ON A QUADRUPEDAL ROBOT

In this section, we evaluate our approach on a quadrupedal robotic platform. With these experiments we aim to demonstrate: 1) Our method is capable of keeping the system safe in a simple do-not-collide task, and 2) Our method can adapt online to measurement uncertainty in different environments without ground-truth data.

A. Hardware System

For the hardware experiments we designed a custom camera array with three equally spaced inexpensive CMOS, global shutter, time-synchronized Arducam cameras. An Nvidia Jetson Nano is used to capture, downsize, and greyscale the stereo images. The images are then sent to an external computer that receives the images and outputs the filtered control input at a frequency of at least 10 Hz. The robot used in this experiment is a Unitree A1

quadrupedal robot that receives inputs of velocity and angle rate, $\mathbf{u} = [v \ \omega]^\top$. A 1 kHz Inverse Dynamics Quadratic Program (ID-QP) walking controller designed using the concepts in [26], is used to track these inputs. Stereo pixel-matching calculations were performed using Efficient Large-scale Stereo (ELAS) [17].

B. Learning Method and Model

The architecture of the model used to estimate \mathcal{P}^θ is a modified version of the Hierarchical Multi-Scale Attention for Semantic Segmentation introduced in [27]; this model is relatively lightweight, consisting of only 196 thousand parameters (e.g., network weights). The robustness threshold used was $\sigma = 0.99$ and the online learning rate was 0.001. We pretrain the model until convergence on a dataset of 6000 stereo image triples collected by manually moving the camera array through a variety of environments.

C. Dynamics Model and Control

In order to control the system we consider a reduced order model of the system dynamics given by the standard unicycle model. The specific form of (2) for this system is:

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{g}(\mathbf{x})} \underbrace{\begin{bmatrix} v \\ \omega \end{bmatrix}}_{\mathbf{k}(\mathbf{x})} \quad (16)$$

A formal analysis of CBFs which utilize reduced-order velocity input models is described in [28].

For this system we consider the pixel-wise CBFs,

$$h(\mathbf{x}, \rho_p) = \frac{1}{2} \left(\left\| \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} \rho_{p,x} \\ \rho_{p,y} \end{bmatrix} \right\|_2^2 - c^2 \right) \quad (17)$$

where $\rho_{p,x}$ and $\rho_{p,y}$ indicate the global real-world x and y positions of pixel p . This function characterizes safety as remaining a planar distance $c > 0$ from ρ_p . This can be thought of as buffering surfaces in the environment by a radius c .

D. Robustness to Uncertainty

To illustrate the efficacy of our method we use two controllers in our experiments. A standard, unrobustified controller:

$$\begin{aligned} \mathbf{k}_{\text{cbf}} &= \underset{\mathbf{u} \in \mathbb{R}^2}{\text{argmin}} \quad \frac{1}{2} \|\mathbf{k}_{\text{des}}(\mathbf{x}) - \mathbf{u}\|_2^2 \\ \text{s.t.} \quad & \underbrace{-[1 \ 0 \ 0]^\top r(p, \hat{d}_p)}_{\hat{h}} v \geq -\gamma(\min_{p \in K} h(\mathbf{x}, \hat{\rho}_p)), \\ & \forall p \in \Lambda \end{aligned} \quad (18)$$

and a robustified controller:

$$\begin{aligned} \mathbf{k}_{\text{cbf}}^* &= \underset{\mathbf{u} \in \mathbb{R}^2}{\text{argmin}} \quad \frac{1}{2} \|\mathbf{k}_{\text{des}}(\mathbf{x}) - \mathbf{u}\|_2^2 \\ \text{s.t.} \quad & -v \geq \frac{-\gamma(\min_{p \in K} h(\mathbf{x}, \rho_p^*))}{[1 \ 0 \ 0]^\top r(p, d_p^*)}, \quad \forall p \in \hat{\Lambda}. \end{aligned} \quad (19)$$

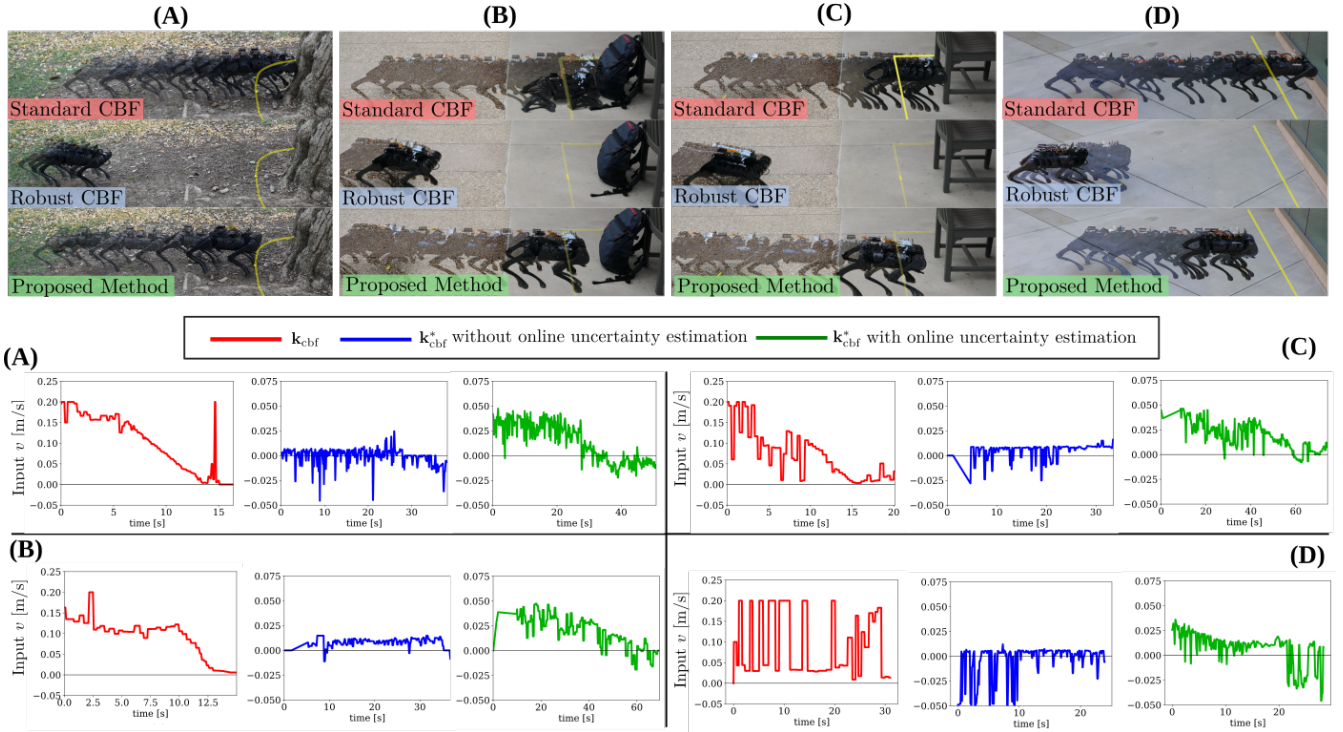


Fig. 4. Demonstration of our method in a variety of environments. From left to right the goal is to maintain a safe distance from (A) a tree, (B) a backpack, (C) a chair, (D) and a glass window. The distance to the barrier is measured and marked on the floor with a yellow tape for visualization purposes – we emphasize this tape is not used for depth estimation. Notice that the barrier is assumed to be a sphere around an obstacle but in the case on the glass, this sphere degenerates into a plane. The quadrupedal robot is given a desired control input of 0.2 m/s. In all cases, a naive barrier implementation that simply takes the noisy measurements from a stereo vision system fails to keep the system safe. The robustified controller (19) with a pretrained model consistently shows overly conservative behavior. Finally, with online learning, the robot converges to the barrier without exhibiting conservative behavior, except for the glass environment where the robot is overly conservative and walks away from the barrier due to the perceived uncertainty. The (A-D) corresponding plots below show the control input filtered by the barrier in each of the three robustification cases.

where $\mathbf{k}_{\text{des}} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ is a desired controller, d^* is the maximum disparity for any $\rho_p \in \mathcal{E}_p$, and ρ_p^* is pixel location associated with d_p^* .

Controller (19) is obtained by first replacing the index set Λ with the $\hat{\Lambda}$. Next we note that $[1 \ 0 \ 0]^\top r(p, \hat{d}_p)$ is strictly positive. After dividing by this quantity, the constraint in (18) is robustified to account for the worst-case error as is done with MR-CBFs. Experimentally, this controller was implemented with $\delta = 0$ and a maximum of 4000 constraints.

E. Experimental Results

The system was run in 4 different environments (see Fig. 4). The CBF (17) was used with a safe radius of $c = 0.33$ m. The intended obstacle in the 4 different environments were (A) a tree, (B) a backpack, (C) a chair, (D) and a glass window. A desired constant forward velocity $v = 0.2$ m/s was used in each experiment and the robot was started approximately 1.3 m away from the obstacle. Since ground-truth measurements were unavailable, we use a yellow line on the ground to indicate the true location of the barrier.

For each environment three different tests were performed. First, controller (18) was used. Since this did not consider measurement uncertainty it failed to achieve safety in every environment; in all experiments the stereo vision overestimated the distance to objects at some point during the run and the quadruped ran directly into the obstacles. Second, the controller (19) was used with an error estimate computed through a pretrained function \mathcal{P}^θ ; this succeeded in providing

safety, but was found to be overly conservative and did not allow the quadruped to approach the obstacle as desired. Third, the controller (19) was used with a \mathcal{P}^θ that adapted to the environment according to Algorithm 2. In this case, safety of the system was generally maintained and over time the system was able to approach the boundary of the safe set. Even when small safety violations occurred, the system eventually corrected and came to rest at a safe steady-state. These results can be seen in Figure 4. A video of the experiments can be found at [29].

V. CONCLUSION AND FUTURE WORK

We presented a framework for achieving safety of a stereo vision-based system using self-supervised online uncertainty estimation and robustified CBFs. Refining the uncertainty estimate model online was shown to achieve significantly better performance. We validated our online learning approach across several environments and successfully achieved robust safety with minimal violations and conservatism.

Future work involves providing mathematical guarantees for our method and extending this theory and application to dynamic environments. We also note that our online uncertainty estimation method, as outlined in Algorithm 2, is a general method that can be coupled with other control or state estimation techniques. Finally, implementing a high-performance on-board version of our hardware system would remove the need for a remote computer.

REFERENCES

- [1] J. C. Knight, "Safety Critical Systems: Challenges and Directions," p. 4.
- [2] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control Barrier Function Based Quadratic Programs for Safety Critical Systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/7782377/>
- [3] Y. Emam, P. Glotfelter, and M. Egerstedt, "Robust barrier functions for a fully autonomous, remotely accessible swarm-robotics testbed," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 3984–3990.
- [4] A. Singletary, K. Klingebiel, J. Bourne, A. Browning, P. Tokumaru, and A. Ames, "Comparative Analysis of Control Barrier Functions and Artificial Potential Fields for Obstacle Avoidance," *arXiv:2010.09819 [cs, eess]*, Oct. 2020, arXiv: 2010.09819. [Online]. Available: <http://arxiv.org/abs/2010.09819>
- [5] P. Glotfelter, J. Cortes, and M. Egerstedt, "Nonsmooth Barrier Functions With Applications to Multi-Robot Systems," *IEEE Control Systems Letters*, vol. 1, no. 2, pp. 310–315, Oct. 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/7937882/>
- [6] N. Cosmay-Shanklin, R. K. Cosner, M. Dai, A. J. Taylor, and A. D. Ames, "Episodic Learning for Safe Bipedal Locomotion with Control Barrier Functions and Projection-to-State Safety," *arXiv:2105.01697 [cs]*, May 2021, arXiv: 2105.01697. [Online]. Available: <http://arxiv.org/abs/2105.01697>
- [7] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.
- [8] S. Dean, N. Matni, B. Recht, and V. Ye, "Robust Guarantees for Perception-Based Control," *arXiv:1907.03680 [cs, math, stat]*, Dec. 2019, arXiv: 1907.03680. [Online]. Available: <http://arxiv.org/abs/1907.03680>
- [9] A. J. Taylor, A. Singletary, Y. Yue, and A. D. Ames, "Learning for Safety-Critical Control with Control Barrier Functions," p. 10.
- [10] M. Ohnishi, L. Wang, G. Notomista, and M. Egerstedt, "Barrier-certified adaptive reinforcement learning with applications to brushbot navigation," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1186–1205, 2019.
- [11] J. Choi, F. Castaneda, C. J. Tomlin, and K. Sreenath, "Reinforcement learning for safety-critical control under model uncertainty, using control Lyapunov functions and control barrier functions," *arXiv preprint arXiv:2004.07584*, 2020.
- [12] E. Hazan, "Introduction to online convex optimization," 2019.
- [13] J. H. Gillula and C. J. Tomlin, "Guaranteed safe online learning via reachability: tracking a ground target using a quadrotor," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 2723–2730.
- [14] B. Sofman, E. Lin, J. A. Bagnell, J. Cole, N. Vandapel, and A. Stentz, "Improving robot navigation through self-supervised online learning," *Journal of Field Robotics*, vol. 23, no. 11–12, pp. 1059–1075, 2006.
- [15] A. Koppel, J. Fink, G. Warnell, E. Stump, and A. Ribeiro, "Online learning for characterizing unknown environments in ground robotic vehicle models," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 626–633.
- [16] M. Perrollaz, A. Spalanzani, and D. Aubert, "Probabilistic representation of the uncertainty of stereo-vision and application to obstacle detection," in *2010 IEEE Intelligent Vehicles Symposium*. IEEE, 2010, pp. 313–318.
- [17] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Asian conference on computer vision*. Springer, 2010, pp. 25–38.
- [18] R. Szeliski and D. Scharstein, "Symmetric sub-pixel stereo matching," in *European Conference on Computer Vision*. Springer, 2002, pp. 525–540.
- [19] K. Liu, K. Ok, W. Vega-Brown, and N. Roy, "Deep Inference for Covariance Estimation: Learning Gaussian Noise Models for State Estimation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, QLD: IEEE, May 2018, pp. 1436–1443. [Online]. Available: <https://ieeexplore.ieee.org/document/8461047/>
- [20] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *53rd IEEE Conference on Decision and Control*. Los Angeles, CA, USA: IEEE, Dec. 2014, pp. 6271–6278. [Online]. Available: <http://ieeexplore.ieee.org/document/7040372/>
- [21] R. K. Cosner, A. W. Singletary, A. J. Taylor, T. G. Molnar, K. L. Bouman, and A. D. Ames, "Measurement-Robust Control Barrier Functions: Certainty in Safety with Uncertainty in State," *arXiv:2104.14030 [cs, eess]*, Apr. 2021, arXiv: 2104.14030. [Online]. Available: <http://arxiv.org/abs/2104.14030>
- [22] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of Control Barrier Functions for Safety Critical Control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015, arXiv: 1612.01554. [Online]. Available: <http://arxiv.org/abs/1612.01554>
- [23] P. Glotfelter, J. Cortes, and M. Egerstedt, "A Nonsmooth Approach to Controller Synthesis for Boolean Specifications," *IEEE Transactions on Automatic Control*, pp. 1–1, 2020.
- [24] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [25] S. Dean, A. J. Taylor, R. K. Cosner, B. Recht, and A. D. Ames, "Guaranteeing Safety of Learned Perception Modules via Measurement-Robust Control Barrier Functions," *arXiv:2010.16001 [cs, eess, math, stat]*, Oct. 2020, arXiv: 2010.16001. [Online]. Available: <http://arxiv.org/abs/2010.16001>
- [26] J. Buchli, M. Kalakrishnan, M. Mistry, P. Pastor, and S. Schaal, "Compliant quadruped locomotion over rough terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 814–820.
- [27] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical multi-scale attention for semantic segmentation," 2020.
- [28] T. G. Molnar, R. K. Cosner, A. W. Singletary, W. Ubellacker, and A. D. Ames, "Model-free safety-critical control for robotic systems," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 944–951, 2022.
- [29] Supplementary video, <https://vimeo.com/605281037>.