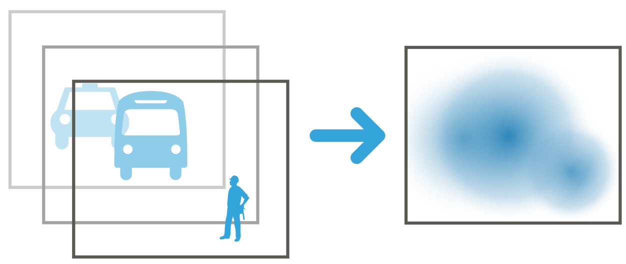
Constant Space Complexity Environment Representation for Vision-based Navigation

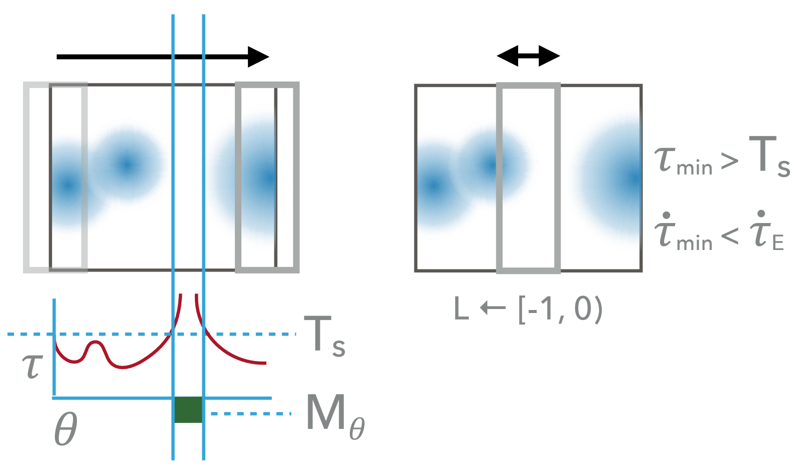
Visualization:



$$F(x,y) = \left\{ \min_{\tau} \left(F_1(x,y), F_2(x,y) \right) \mid (x,y) \in I \right\}$$

Left: Perception and tracking in the image plane output multiple objects

Right: The potential field collapses these objects to a fixed-size representation



Left: Directional control can be determined by convolution of the ISP

Right: Longitudinal control can be determined similarly

Algorithm & Complexity:

In Algorithm 1, all non-trivial operations are iterations over the width of the image plane. The operations on Lines 5 & 7 depend on the user defined parameters, but these are also bounded by image width. In Algorithm 2, Line 4 is a call to Algorithm 1, and Line 11 is assumed to be implemented with an O(C) proportional law. Thus, the algorithm as a whole has constant complexity in space and time, with respect to the camera image space.

Algorithm 1 Given an image space potential field F, compute the set of steering and acceleration commands that satisfy $\tau \geq T_s$ and $\dot{\tau} \geq -0.5 + \varepsilon$, where $T_s > 0$ is some desired time headway, w_θ and w_a are kernel widths for computing steering angle and acceleration maps, and $\varepsilon > 0$ is a buffer.

1: **procedure** SAFECONTROLS($F, T_s, \dot{\tau}_E, w_\theta, w_a, \varepsilon$)
2: Let I_c be the list of image column indices
3: Let M_a map $i \in I_c$ to steering angles
4: Let h be the height (row count) of F
5: Let M_τ map $\langle \tau, \dot{\tau} \rangle$ to $i \in I_c$ via $w_\theta \times h$ min filter
6: Let $M_\theta = \{ \langle \tau, \dot{\tau} \rangle \in M_\tau \, : \, \tau \geq T_s \}$
7: Let W be a centered $w_a \times h$ window in F
8: Let $\langle \tau, \dot{\tau} \rangle_{\min}$ be the min. τ over W
9: Let $L \leftarrow \emptyset$ be a container for safe accelerations
10: **if** $M_\theta = \emptyset$ **then**
11: $M_\theta \leftarrow 0$, $L \leftarrow [-1, -1]$
12: **else if** $\tau_{\min} > T_s$ **then**
13: $L \leftarrow [-1, 1]$
14: **else**
15: **if** $f(\dot{\tau}, \varepsilon) = 0$ **then**
16: $L \leftarrow [-1, -1]$
17: **else**
18: $L \leftarrow [-1, 0)$
19: **end if**
20: **end if**
21: **return** M_θ, L
22: **end procedure**

Algorithm 2 For a desired pixel location (x_d, y_d), and setpoint speed $\dot{s}_d$, compute the Selective Determinism control that safely guides the agent A toward (x_d, y_d). See Algorithm 1 for descriptions of the other parameters.

1: **procedure** CONTROLS($(x_d, y_d), F, T_s, \dot{\tau}_E, w_\theta, w_a, \varepsilon$)
2: Let $\theta_t, \dot{s}_t$ be the steering angle and speed of A
3: Let θ_d be the steering angle corresponding to y_d
4: Let $M_\theta, L \leftarrow$ SafeControls($F, T_s, \dot{\tau}_E, w_\theta, w_a, \varepsilon$)
5: Let $\theta^\star \leftarrow \theta_t$ contain the new steering angle
6: **for** $\theta \in M_\theta$ **do**
7: **if** $|\theta - \theta_d| < |\theta^\star - \theta_d|$ **then**
8: $\theta^\star \leftarrow \theta$
9: **end if**
10: **end for**
11: Let $\dot{s}^\star \in L$ be chosen proportionally to $\dot{s}_d - \dot{s}_t$
12: **return** $\theta^\star$, $\dot{s}^\star$
13: **end procedure**

Video:

https://youtu.be/yHoR3ZpX1KE

A visualization of ISP fields can be seen online by scanning the QR code or visiting the link below the QR code.

Jeffrey Kane Johnson

maeve AUTOMATION