Planning High-speed Safe Trajectories in Confidence-rich Maps

Eric Heiden[1], Karol Hausman[1],

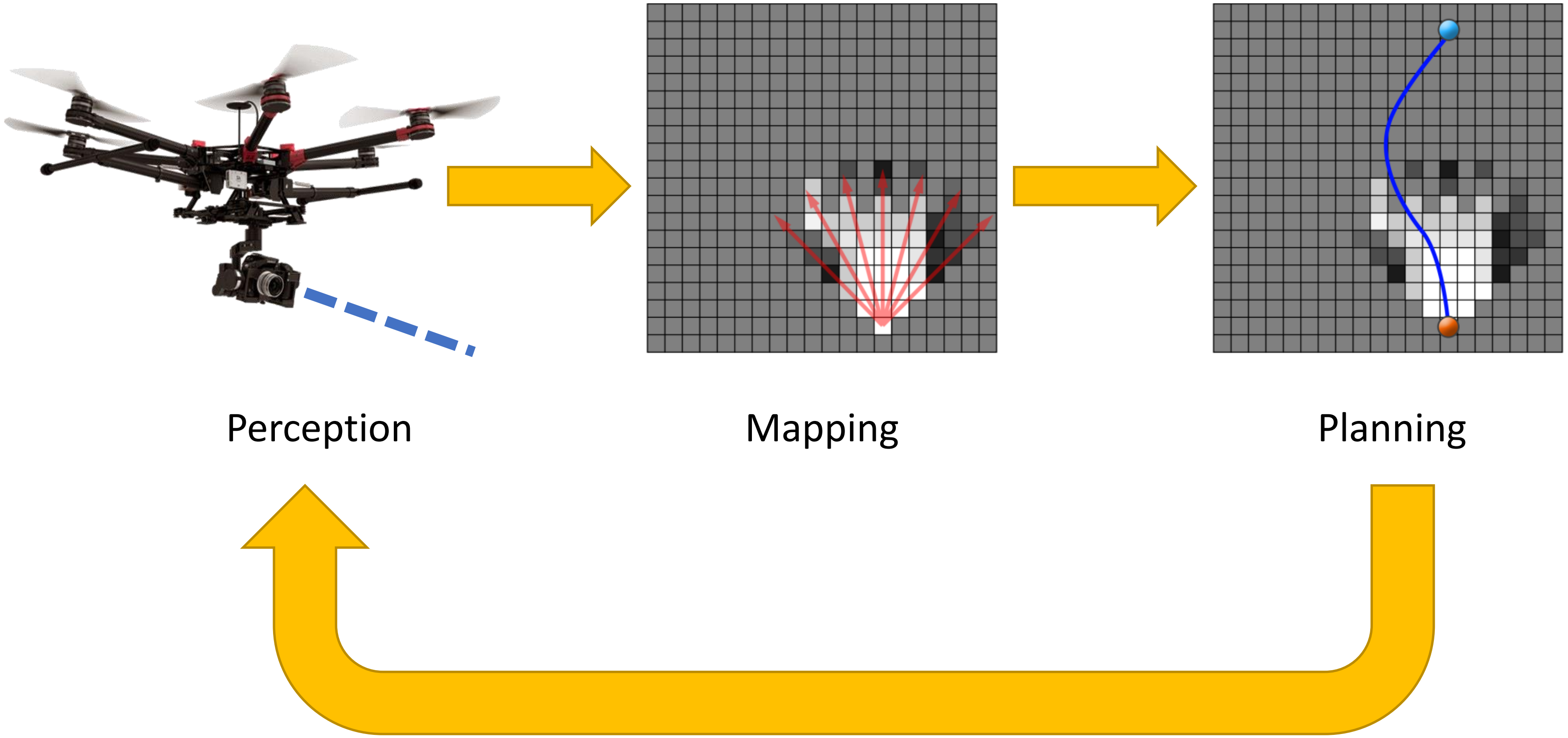
Gaurav S. Sukhatme[1], Ali-akbar Agha-mohammadi[2]

[1] University of Southern California, Los Angeles, USA

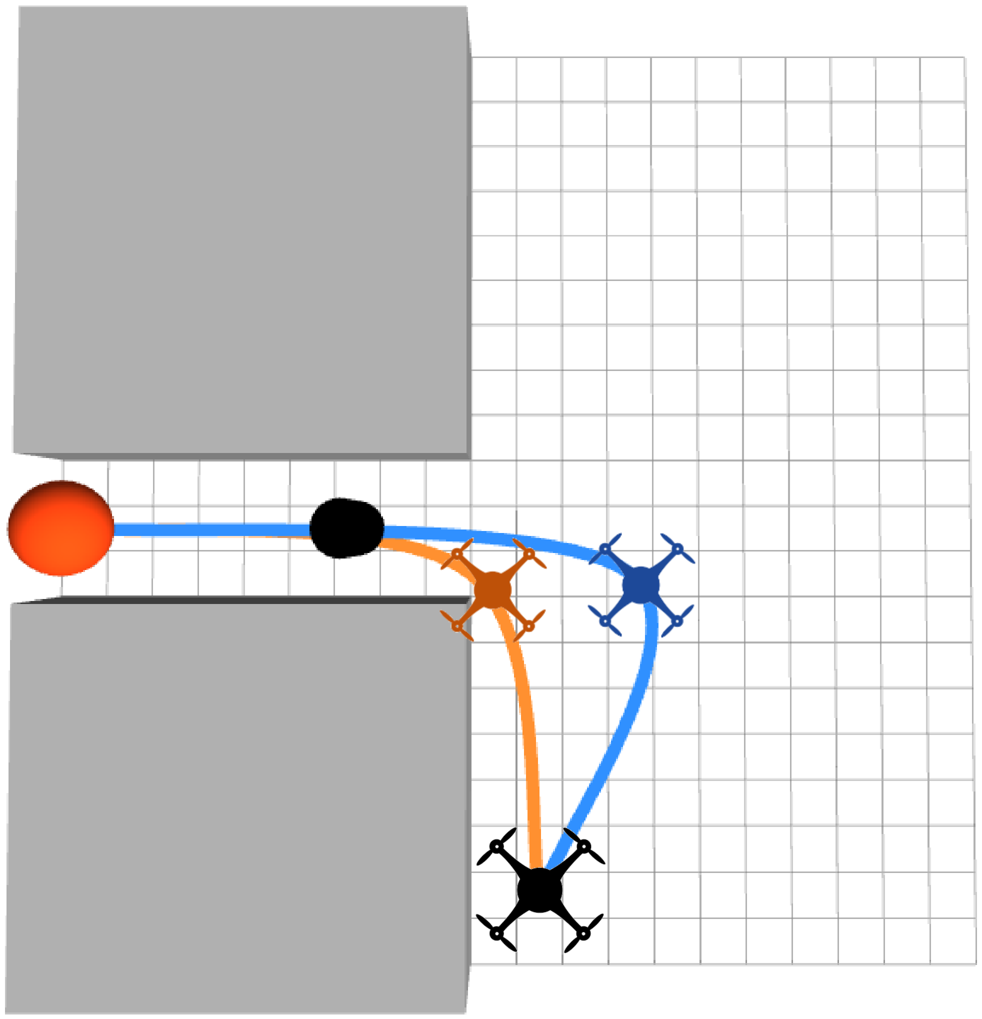
[2] Jet Propulsion Laboratory, California Institute of Technology
Pasadena, USA

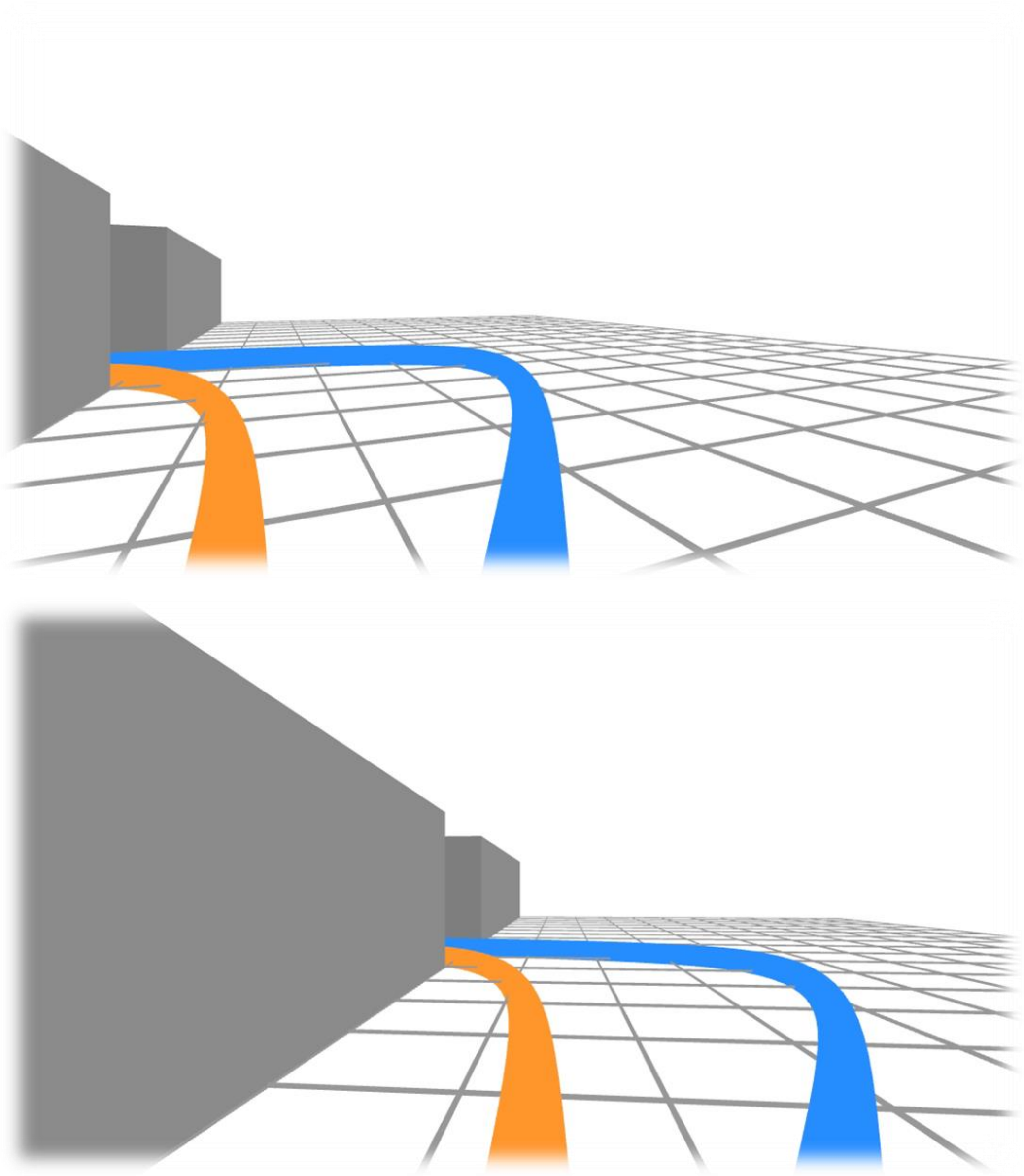
IROS 2017

Simultaneous Mapping And Planning (SMAP)



Reducing "Surprise"

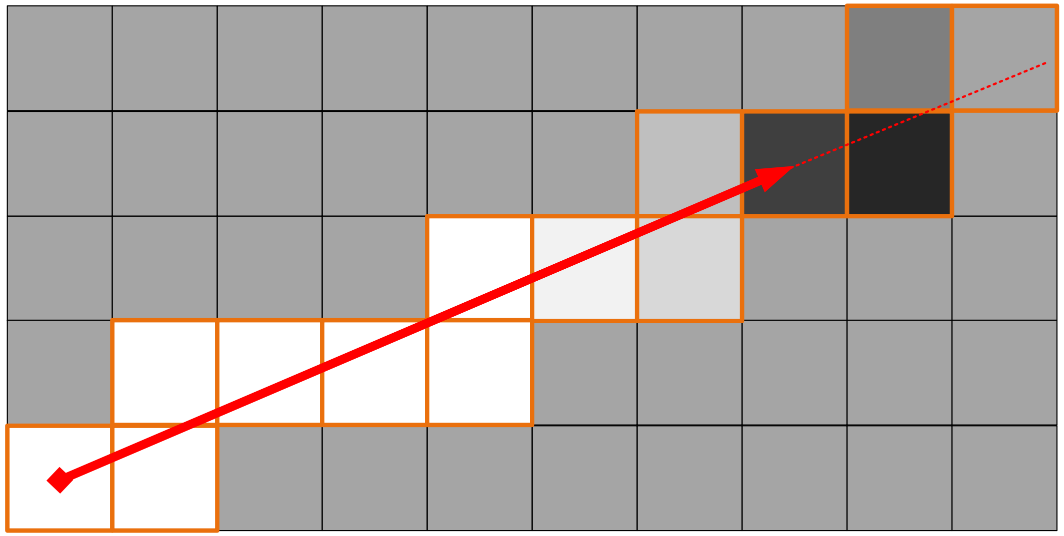




Traditional Occupancy Grids

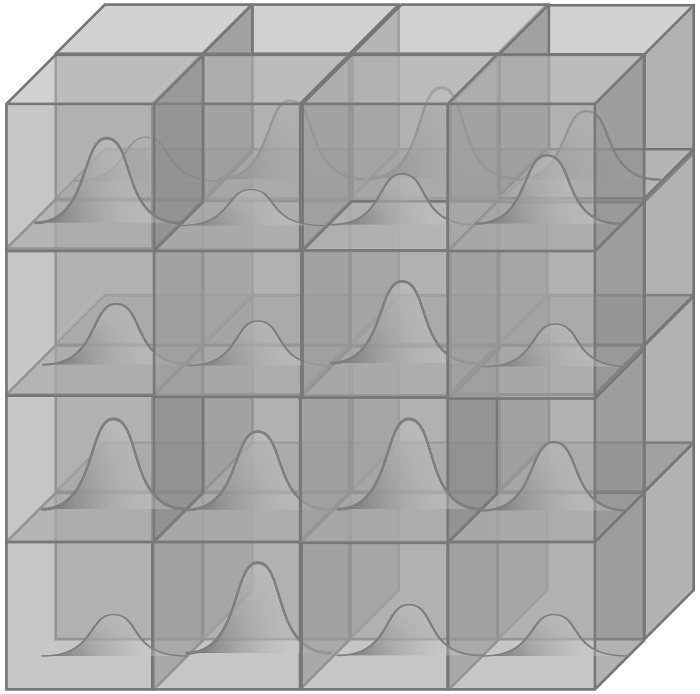
- Incorrect assumption of voxel independence
- Lack of reliable *confidence* measure

➢ Unreliable for planning with noisy sensors



Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, *34*(3), 189–206.

Confidence-rich Maps

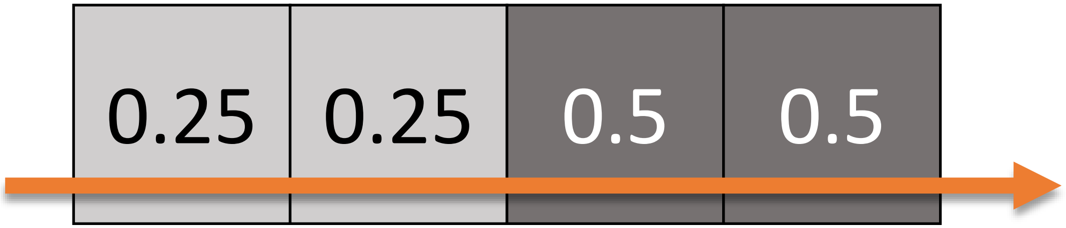


- Stores **pdf** of occupancy
- Interdependence of voxels is maintained

Ali-akbar Agha-mohammadi, Eric Heiden, Karol Hausman, Gaurav Sukhatme: Confidence-rich Grid Mapping.
To Appear in International Symposium on Robotics Research (ISRR), 2017.

Utility Function for Safe Planning

Reachability



$$R = \prod_{i=1}^{n}(1 - m_i)$$

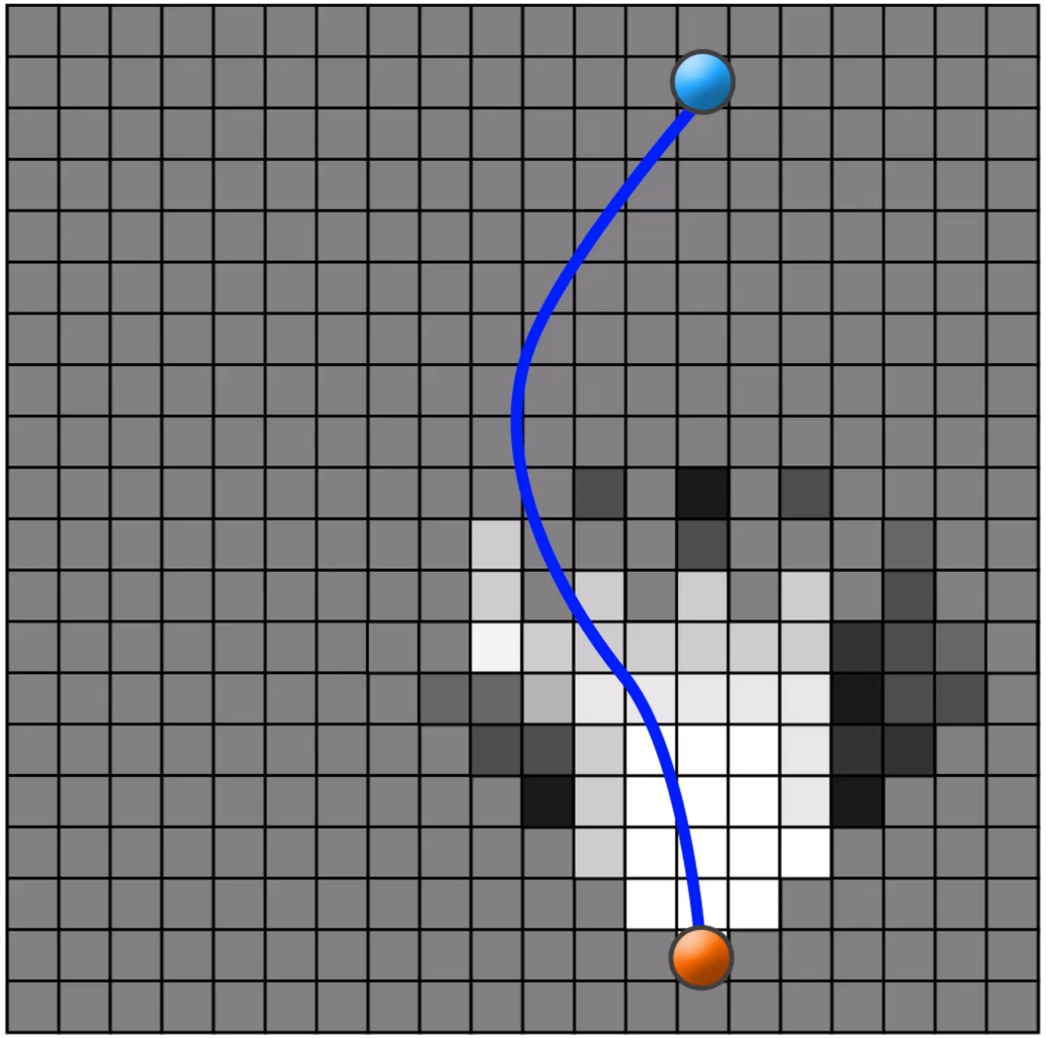
Lower Confidence Bound

Utility function to combine expected reachability and confidence

$$\mathrm{LCB} = \mathbb{E}[R(\mathbf{x})] - \kappa\sigma[R(\mathbf{x})]$$

SMAP Algorithm

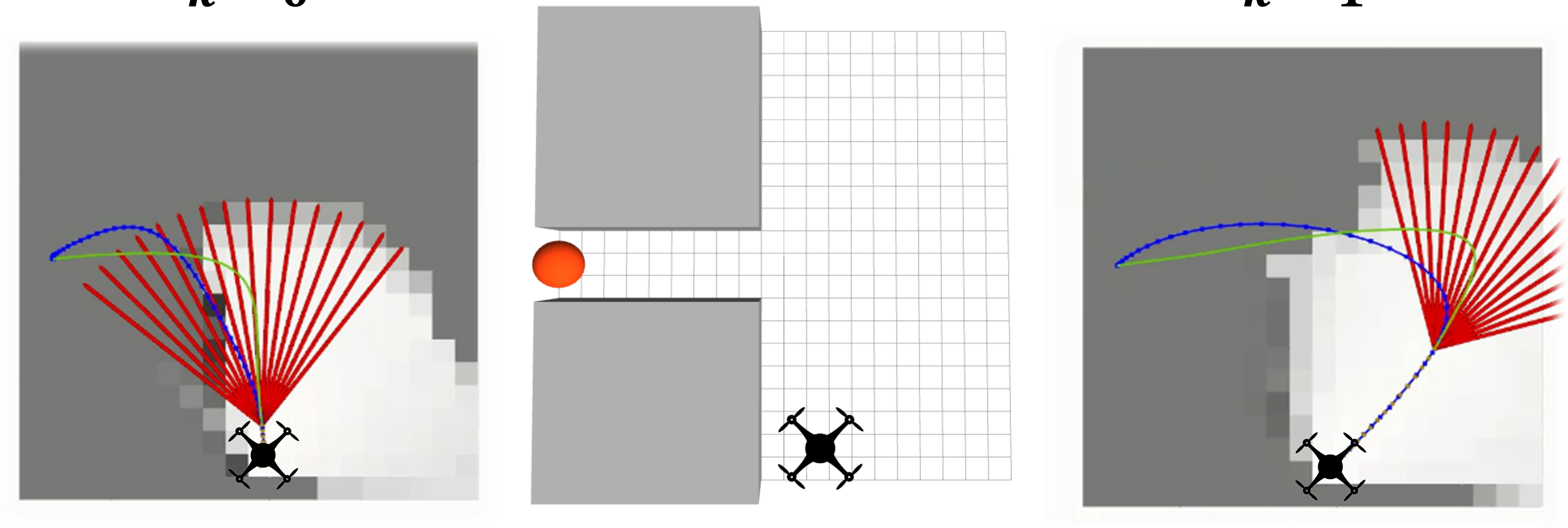
Compute trajectory $\mathbf{x}$
while $t < T$ **do**
    Set pose to $\mathbf{x}(t)$
    Observe, update map belief b^m
    $r_t \leftarrow R\big(\mathbf{x}(t\colon t + k\Delta t)\big)$
    if $r_t < 1 - \epsilon$ **then**
        $\mathbf{x} \leftarrow \arg\max_{\mathbf{x}'} \mathrm{LCB}(\mathbf{x}', b^m)$
    end if
    $t \leftarrow t + \Delta t$
end while



Performance w.r.t $\boldsymbol{\kappa}$

$$\mathrm{LCB} = \mathbb{E}[R(\mathbf{x})] - \kappa\sigma[R(\mathbf{x})]$$

$\boldsymbol{\kappa = 0}$ $\boldsymbol{\kappa = 1}$

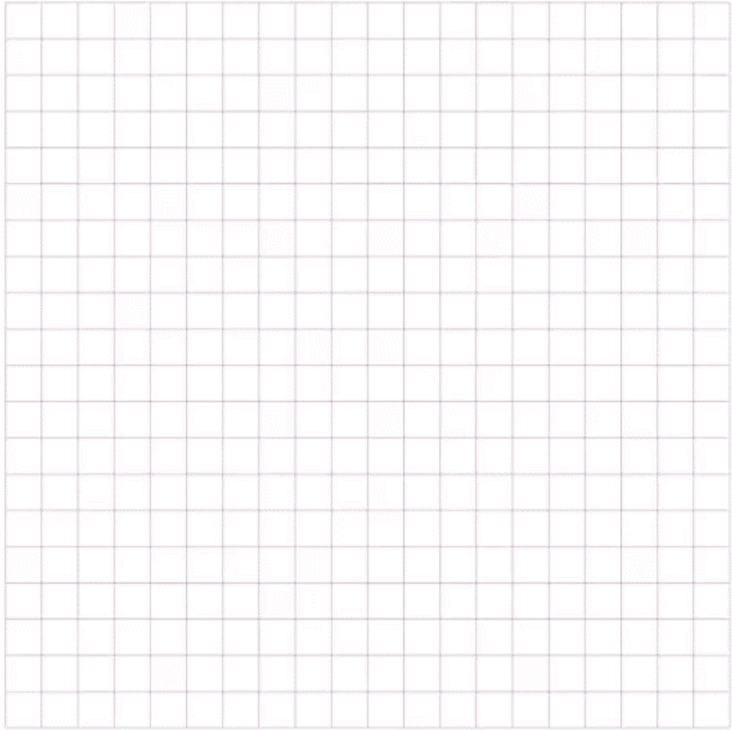


Current trajectory $\mathbf{x}(t)$ Optimization candidate $\mathbf{x}'(t)$

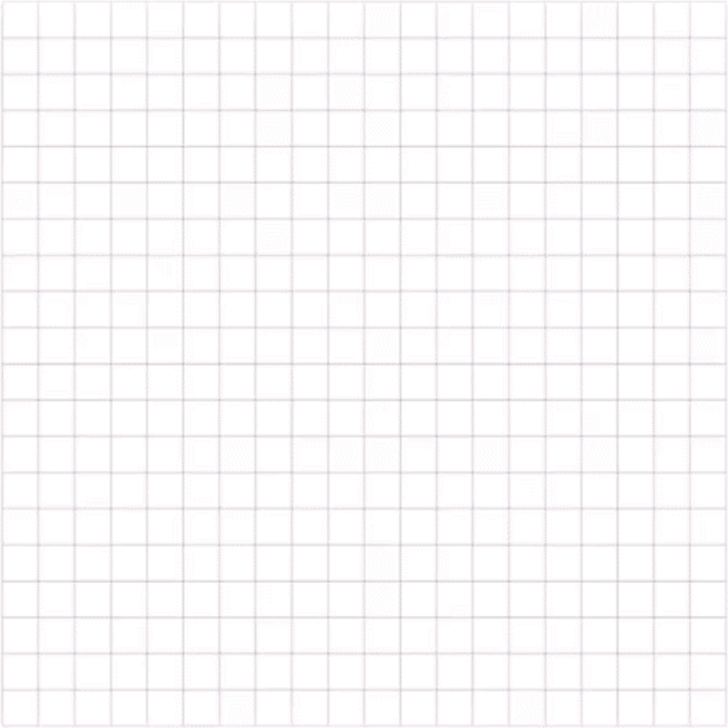
Performance w.r.t κ

$\mathrm{LCB} = \mathbb{E}[R(\mathbf{x})] - \kappa\sigma[R(\mathbf{x})]$

$\boldsymbol{\kappa} = \mathbf{0}$

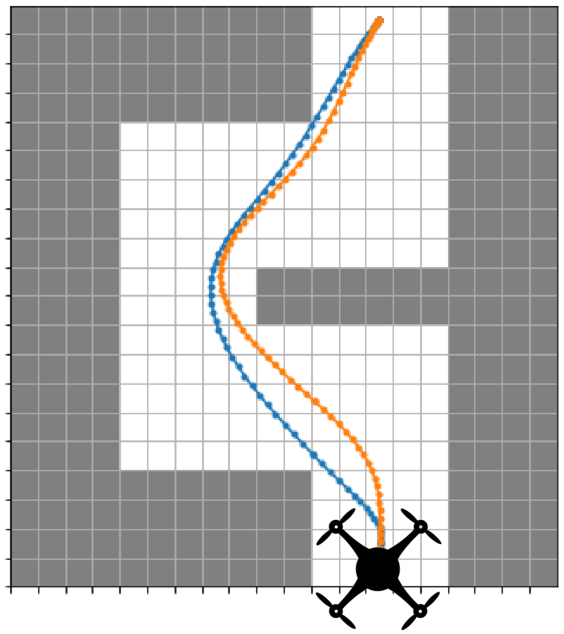


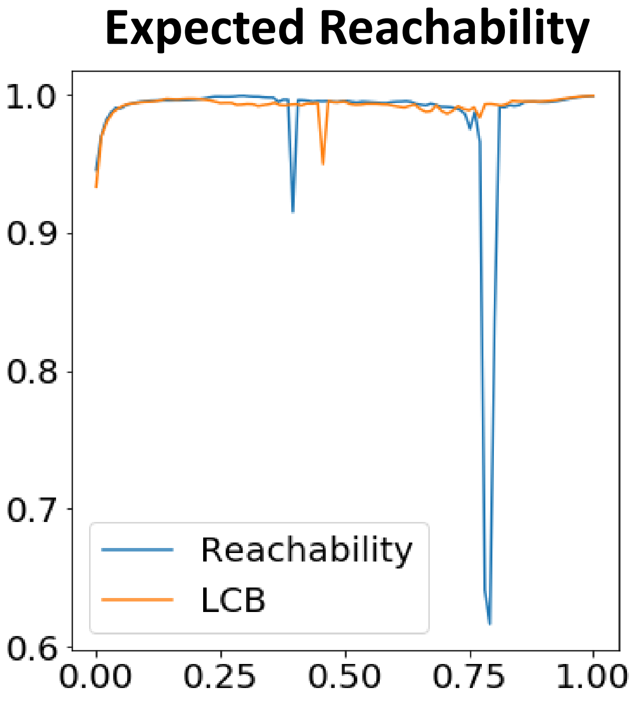
$\boldsymbol{\kappa} = \mathbf{1}$

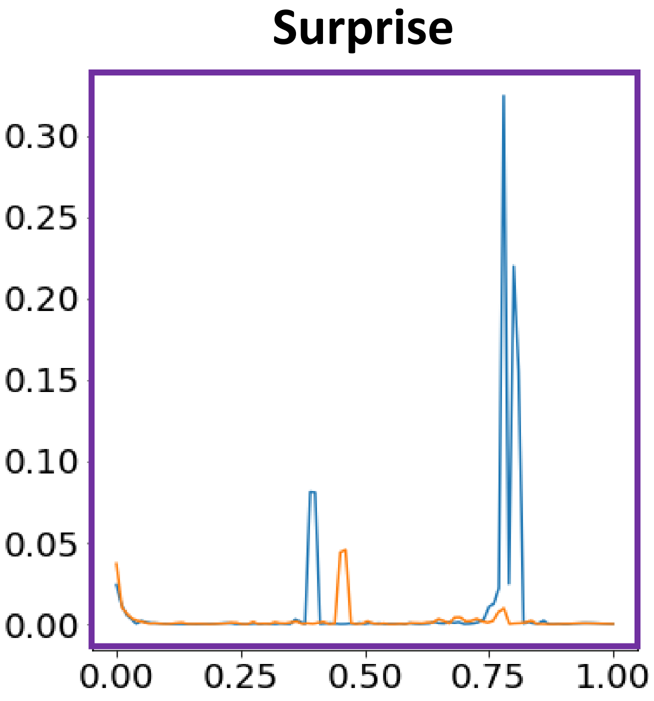


Current trajectory $\mathbf{x}(t)$ Optimization candidate $\mathbf{x}'(t)$

Results







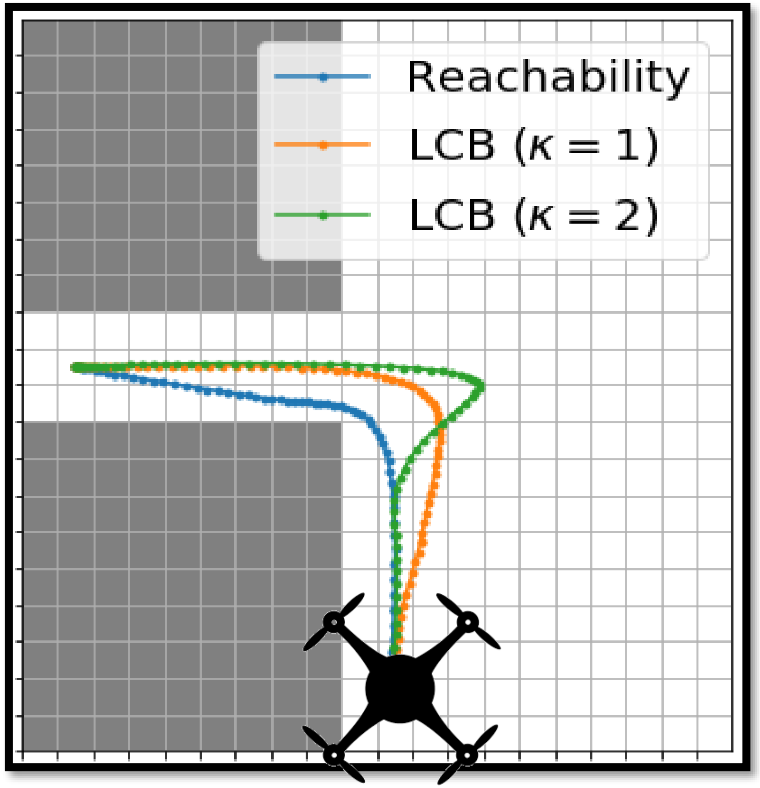
(LCB with $\kappa = 1$)

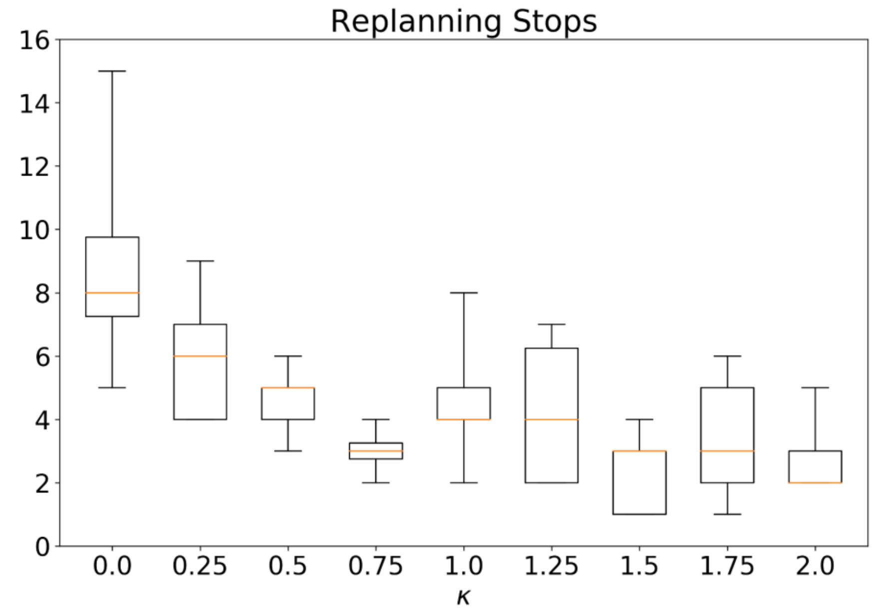
Strong reduction in ***surprise***:

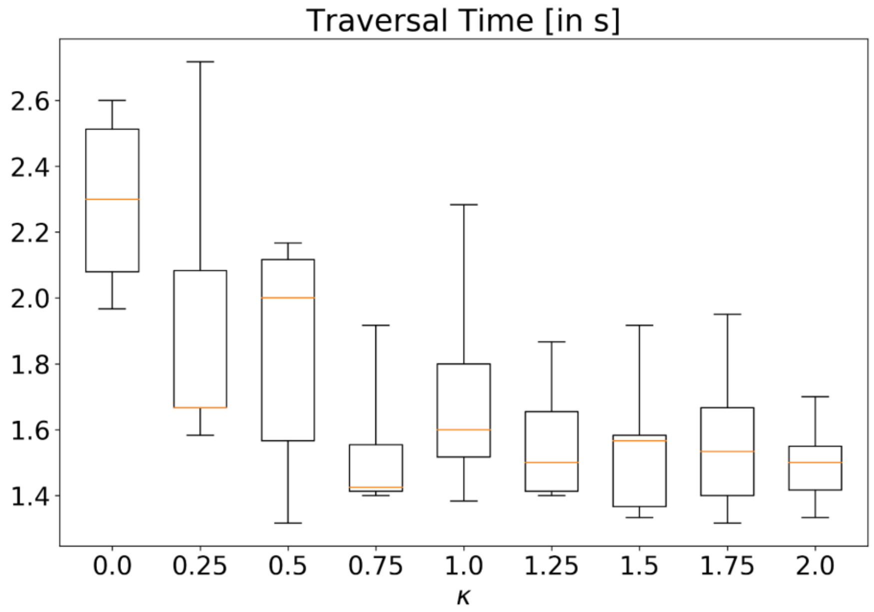
$$|\Delta R(\mathbf{x}(t))| = |R(\mathbf{x}(t)) - R(\mathbf{x}(t - \Delta t))|$$

Results

$$\text{LCB} = \mathbb{E}[R(\mathbf{x})] - \boldsymbol{\kappa}\sigma[R(\mathbf{x})]$$







Reduction in re-planning stops up to 70%

Speed-up in traversal time up to 34%

Planning High-speed Safe Trajectories in Confidence-rich Maps

Eric Heiden[1], Karol Hausman[1], Gaurav S. Sukhatme[1], Ali-akbar Agha-mohammadi[2]

Contributions:

- Incorporated novel mapping approach into planning pipeline
- Analyzed probabilistic safety measure for a trajectory
- Proposed a utility function that leverages covariance of map belief

[1] University of Southern California, Los Angeles, USA

[2] Jet Propulsion Laboratory, California Institute of Technology Pasadena, USA

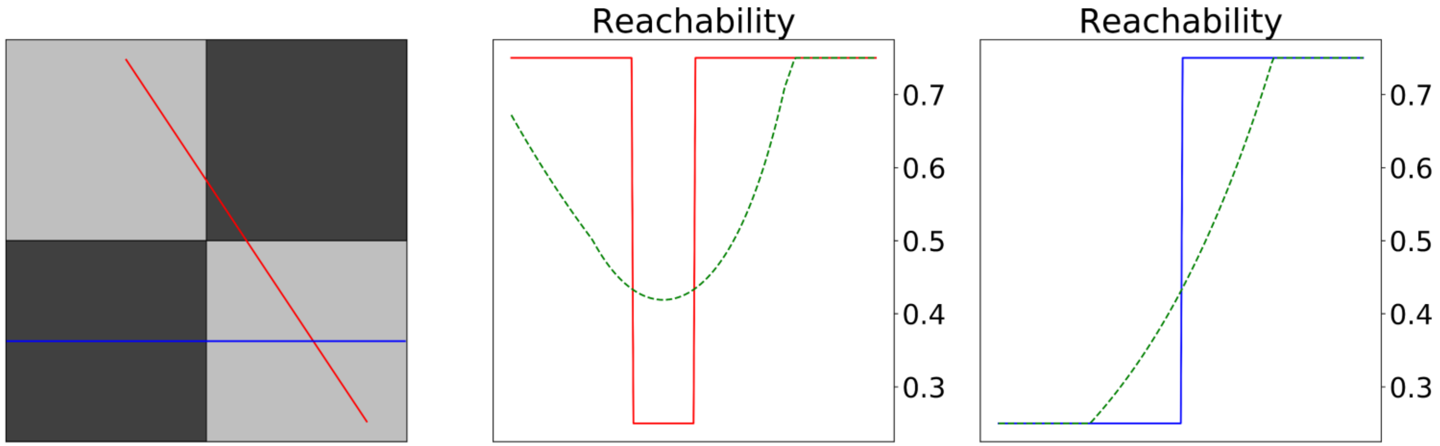
IROS 2017

Planning Methods

- Global, sampling-based planning in binary grid maps: RRT*, PRM*, Theta*, etc.
- Re-planning: D*
- Trajectory optimization: CHOMP, TrajOpt, etc.
- Global planning in GP maps: functional gradients
- Learning: guidance functions, cognitive mapping and planning, visual navigation, etc.

Reachability as Random Variable

Product of Random Variables



$$\mathbb{E}[R(x)] = \lim_{\Delta \to 0} \prod_{i=0}^{n} (\mathbf{1} - \boldsymbol{m}[\boldsymbol{x}(\boldsymbol{t_i})])^{d_i} \quad \text{where } d_i = \|x(t_i) - x(t_i + \Delta t)\|$$

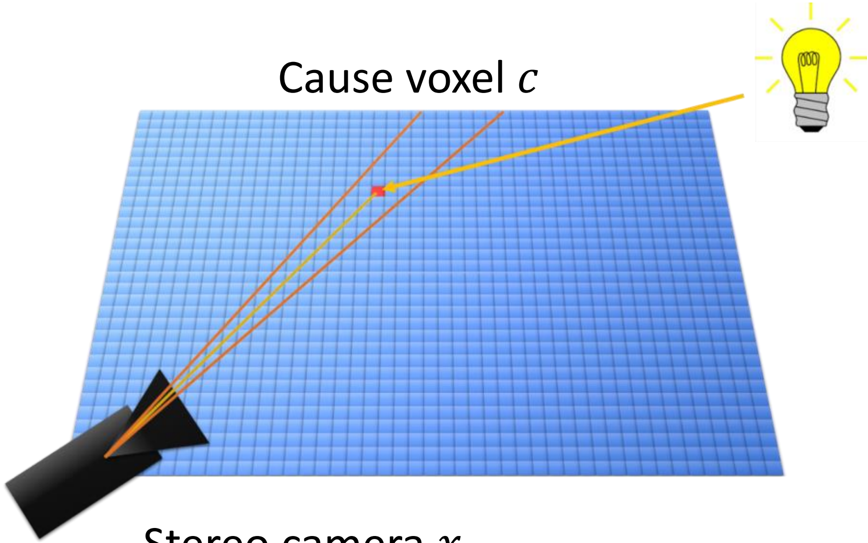
Compute $\mathrm{Var}[R(x)]$ accordingly using variance of occupancy estimates.

Confidence-rich Maps

Which voxel was cause of the measurement?

$$p(c|b^m) = \Pr(B^c, R^c|b^m)$$

Light has to bounce off voxel c and it has to reach the camera.



Sensor Cause Model

$$\forall c_k \in \mathbb{C}(x_k):\ p(c_k|z_{0:k}, x_{o:k}) = p(c_k|b^m_{k-1}, z_k, x_k)$$

$$= \eta' p(z_k|c_k, x_k)\hat{m}^{c_k}_{k-1} \prod_{j=1}^{c^l_k - 1} \left(1 - \hat{m}^{g(j,x)}_{k-1}\right)$$

Every voxel stores pdf of $p\left(m^i \middle| z_{0:k}, x_{0:k}\right)$.

Trajectory Generation

Minimum Snap trajectory generation

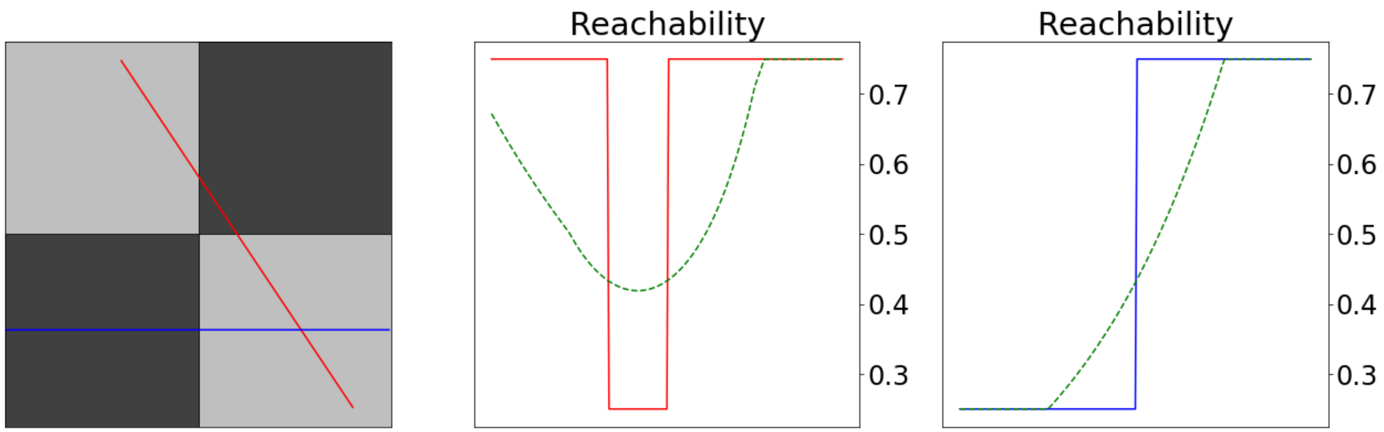
$$\mathbf{x}(t) = \begin{cases} P_1\mathbf{t}(t) & \text{if } t_0 \le t < t_1 \\ \vdots & \\ P_q\mathbf{t}(t) & \text{if } t_{q-1} \le t \le t_q \end{cases}$$

Parameterize polynomial via weights on null-space of time matrix $\mathbf{T}$ satisfying constraints $\mathbf{c}$

$$\bar{\mathbf{P}}^* = \mathbf{c}\mathbf{T}^+ + \mathbf{r}Null(\mathbf{T}^T)^T$$

Mellinger, Daniel, and Vijay Kumar. "Minimum snap trajectory generation and control for quadrotors.“ *International Conference on Robotics and Automation (ICRA)*. IEEE, 2011.

Reachability as a Random Variable



(a) (b) (c)

$$\mathbb{E}[R(x)] = \lim_{\Delta \to 0} \prod_{i=0}^{n} (\mathfrak{r}[x(t_i)])^{d_i}$$

$$\text{var}[R(x)] = \lim_{\Delta \to 0} \prod_{i=0}^{n} (\hat{v}[x(t_i)] + (\mathfrak{r}[x(t_i)])^2)^{d_i} - \prod_{i=0}^{n} (\mathfrak{r}[x(t_i)])^{2d_i}$$

where $d_i = \|x(t_i) - x(t_i + \Delta t)\|$ and $\mathfrak{r}(x) = 1 - m(x)$.

Bilinear Interpolation

Compute reachability at continuous position $x(t)$ in a $U \times V$ voxel map:

$$\mathfrak{r}[\mathbf{x}(t)] \approx \left((1 - m_{ij})^{1-\alpha} \cdot (1 - m_{i+1,j})^{\alpha}\right)^{1-\beta} \cdot \left((1 - m_{i,j+1})^{1-\alpha} \cdot (1 - m_{i+1,j+1})^{\alpha}\right)^{\beta},$$

where:

$$i = \lfloor x_t U \rfloor \qquad j = \lfloor y_t V \rfloor$$
$$\alpha = \mathrm{frac}(x_t U) \qquad \beta = \mathrm{frac}(y_t V).$$

Re-planning Algorithm

Compute trajectory x
while $t < T$ **do**
    Move to position $x(t)$ forward-facing
    Observe, update map belief b^m
    $r_t \leftarrow R\big(x(t : t + k\Delta t)\big)$
    if $r_t < 1 - \epsilon$ **then**
        $x \leftarrow \arg\max_x \text{Evaluate}(x, t, b^m)$
    end if
    $t \leftarrow t + \Delta t$
end while

procedure $\text{Evaluate}(x, t, b^m)$**:**
$t' \leftarrow t + \Delta t$
$b^{m,ml} \leftarrow b^m$
while $t' < T$ **do**
    Move to $x(t)$ forward-facing
    Observe most-likely measurement
    $z^{ml} = \arg\max_z p\big(z \big| b^{m,ml}, xv\big)$
    Update map belief $b^{m,ml}$
    $d \leftarrow ||\mathbf{x}(t_i) - \mathbf{x}(t_i + \Delta t)||$
    $\mathcal{R} \leftarrow R \cdot (\mathfrak{r}[\mathbf{x}(t)])^d$
    $\mathcal{V}_l \leftarrow \mathcal{V}_l \cdot \left(v(\mathbf{x}(t)) + (\mathfrak{r}[\mathbf{x}(t)])^2\right)^d$
    $\mathcal{V}_r \leftarrow \mathcal{V}_r \cdot (\mathfrak{r}[\mathbf{x}(t)])^{2d}$
    $t' \leftarrow t' + \Delta t$
end while
$v \leftarrow v_l - v_r$
$LCB \leftarrow R - \kappa\sqrt{v}$
return LCB