Lecture: Autonomous micro aerial vehicles

Friedrich Fraundorfer

Remote Sensing Technology

TU München

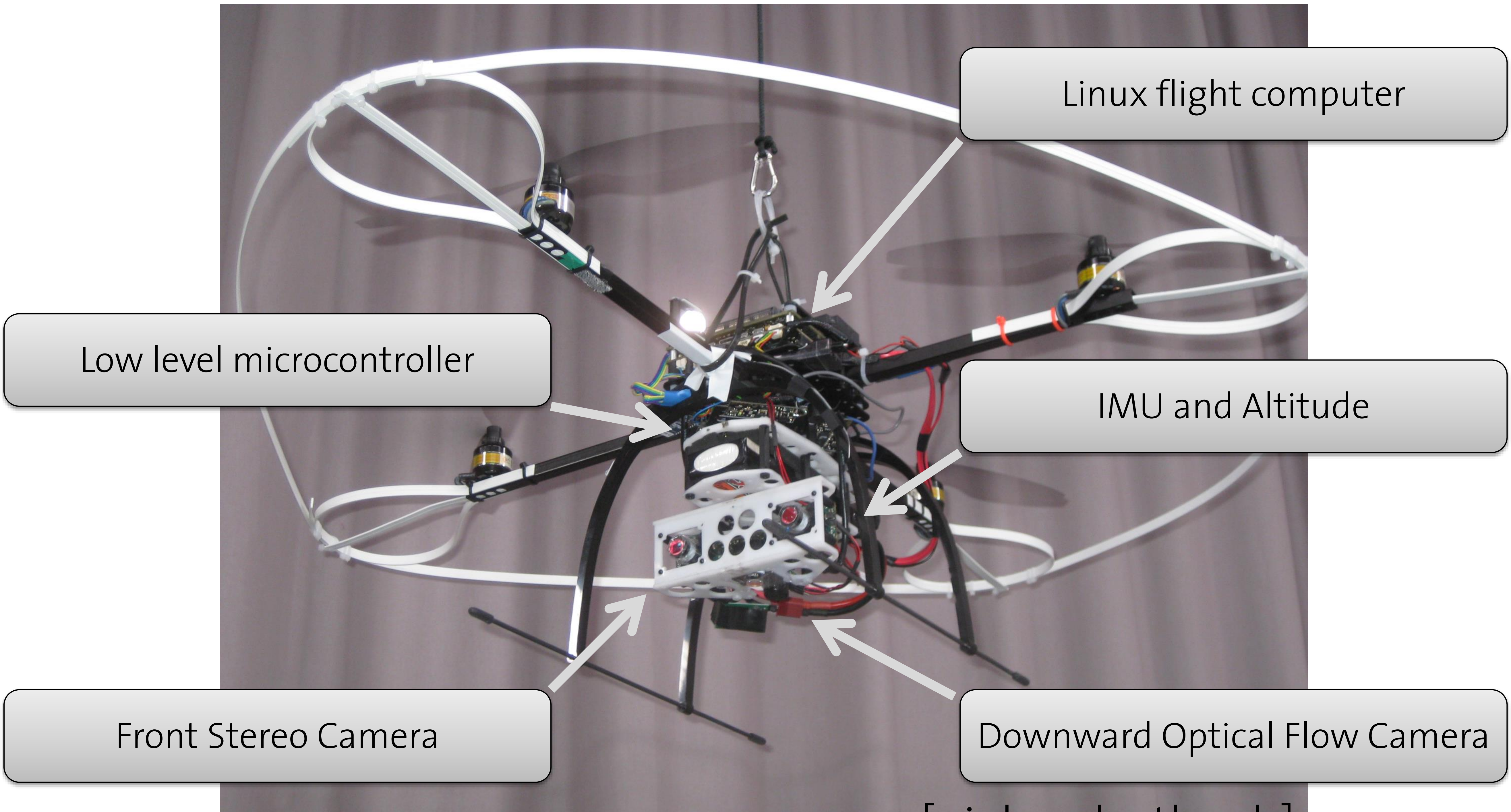
Autonomous operation@ETH Zürich

Autonomous operation@ETH Zürich

Outline

- MAV system
- Optical flow for pose estimation
- 3D mapping for navigation
- Navigation
  - Frontier based exploration
  - VFH+/State lattice

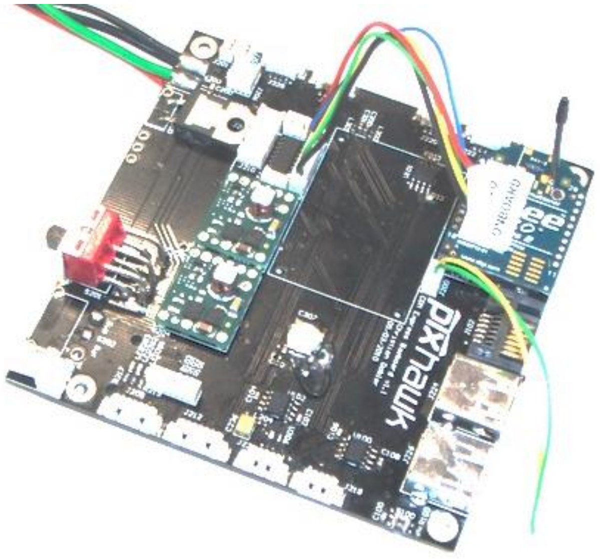
Sensor and System Setup

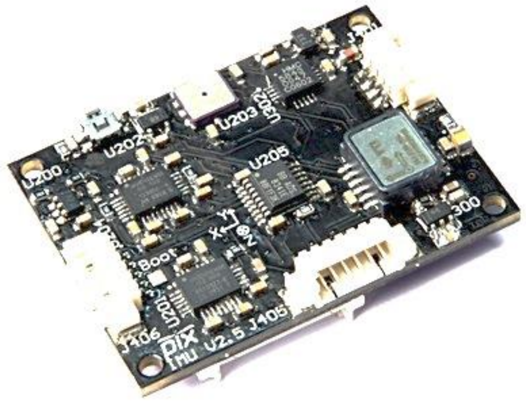


[pixhawk.ethz.ch]

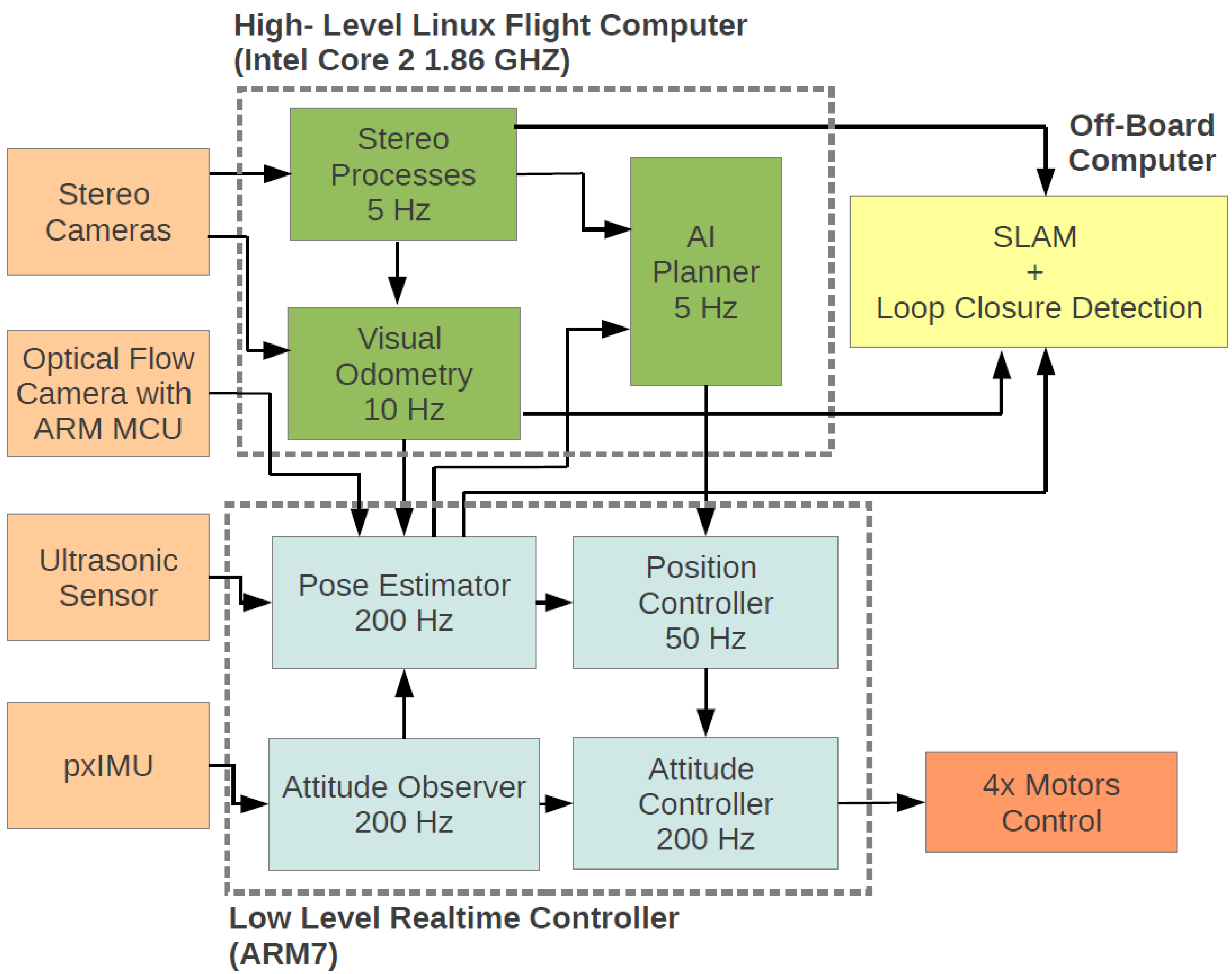
Onboard Electronics

- Computer-On-Module (COTS, 18-27W)
  - microETXexpress industry standard module
  - Intel Core 2 DUO 1.86 GHz / Core i7 2.0 GHz (18-40 W)
- Onboard flight computer (pxCOMex)
  - PIXHAWK mainboard design
  - 200g (incl. cooling, 27W Core 2, 40W Core i7)
- Inertial Measurement Unit (pxIMUv2.5)
  - 60 MHz ARM7
  - 3D gyroscope, accelerometer, compass
  - Barometric pressure
  - Kalman filtering and PID controllers
  - Triggers cameras

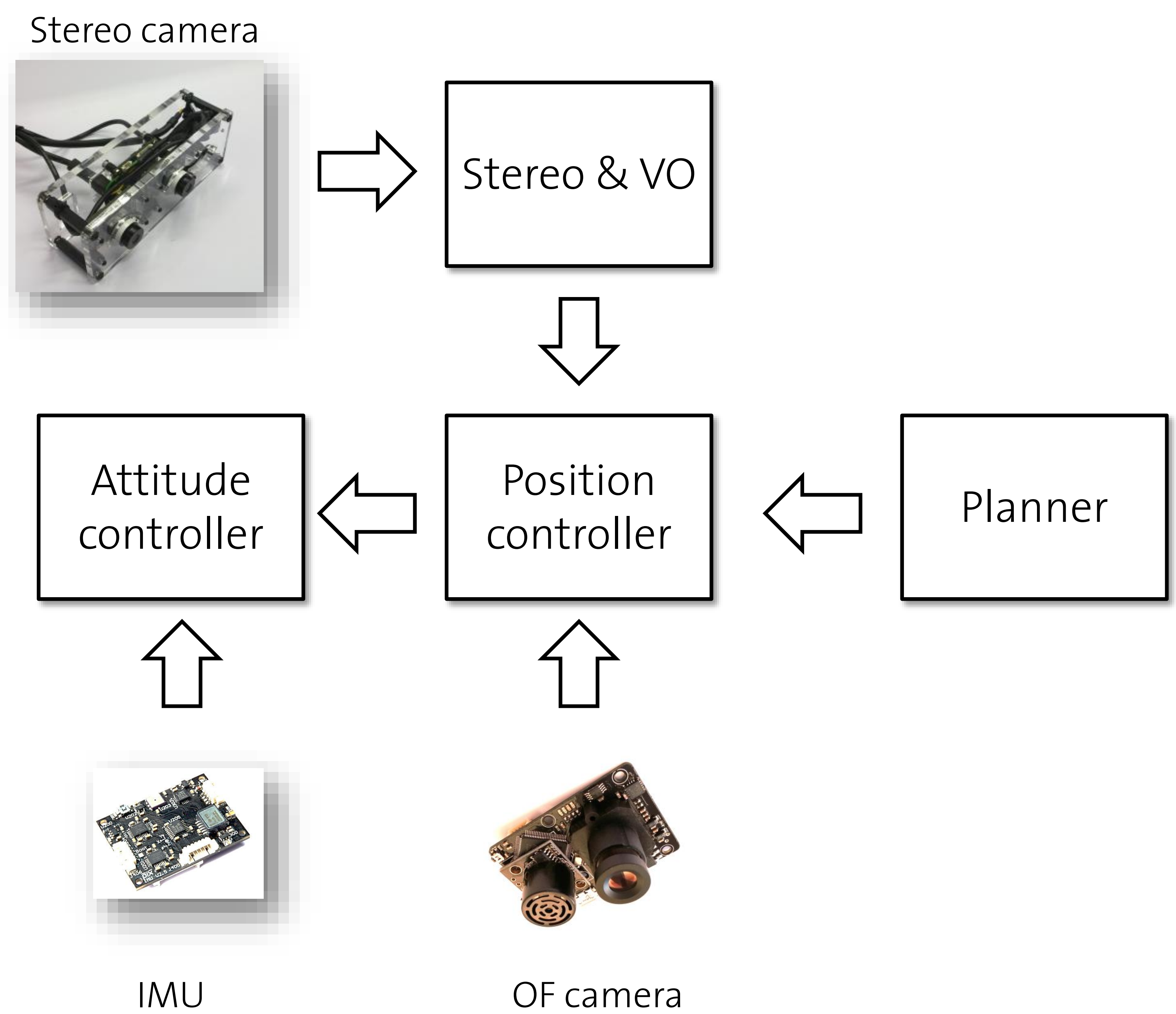




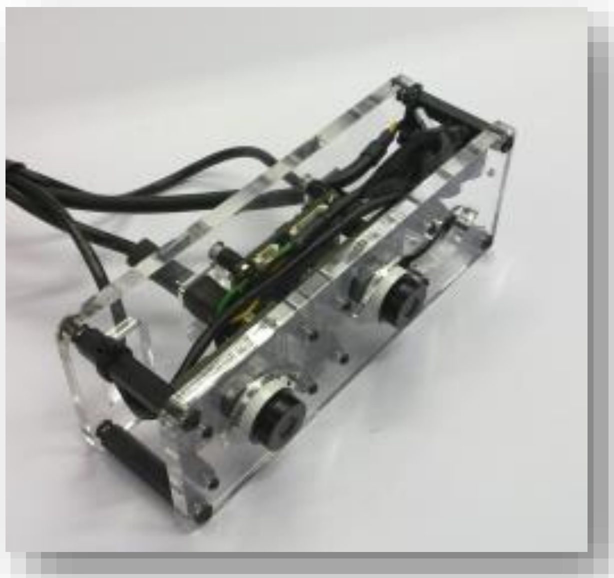
System Architecture



Autonomous Flight



Important sensors



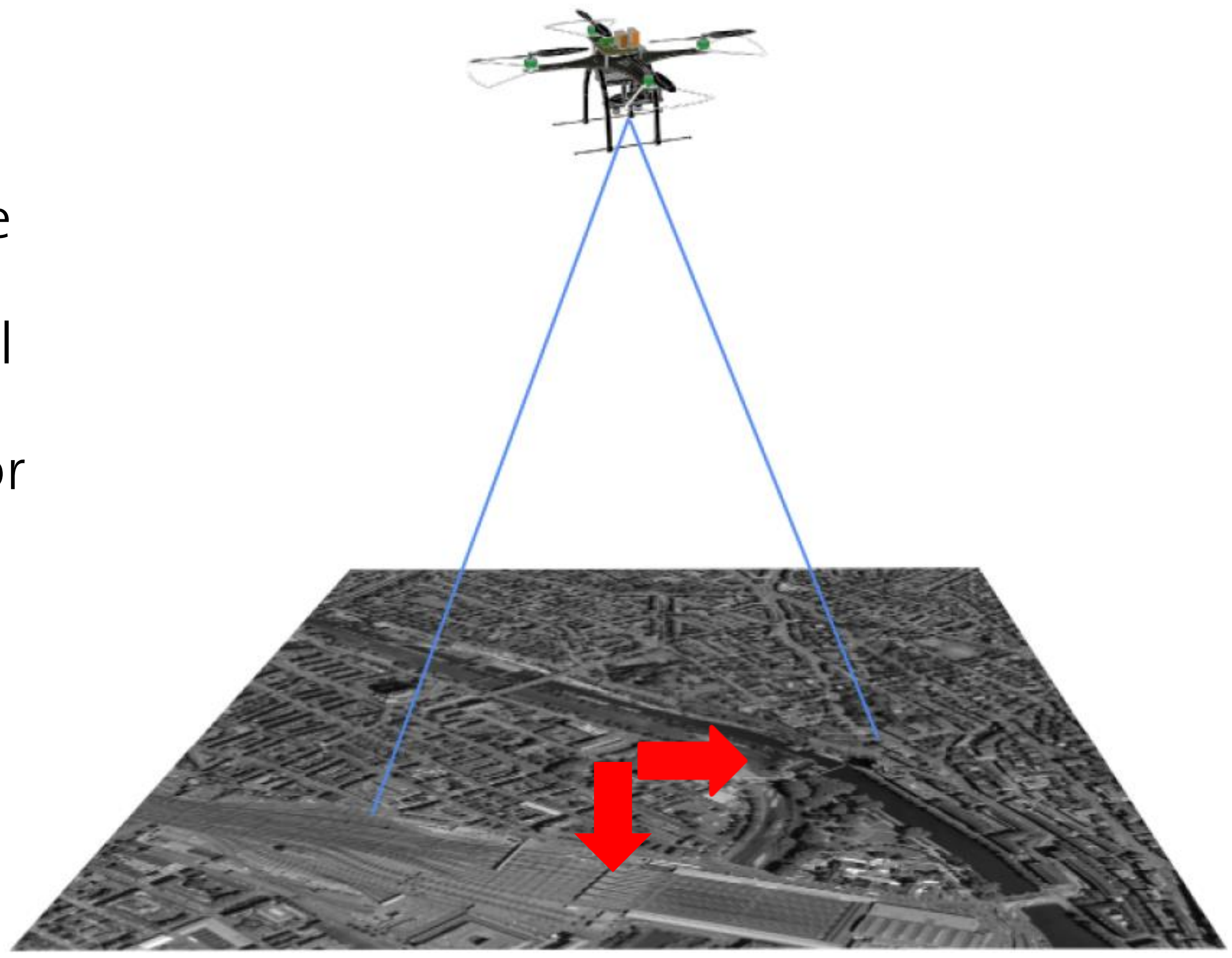
- Stereo camera (timestamped and synchronized with IMU measurements, max. 30fps)

- Optical flow camera

The optical flow idea

Assumptions:

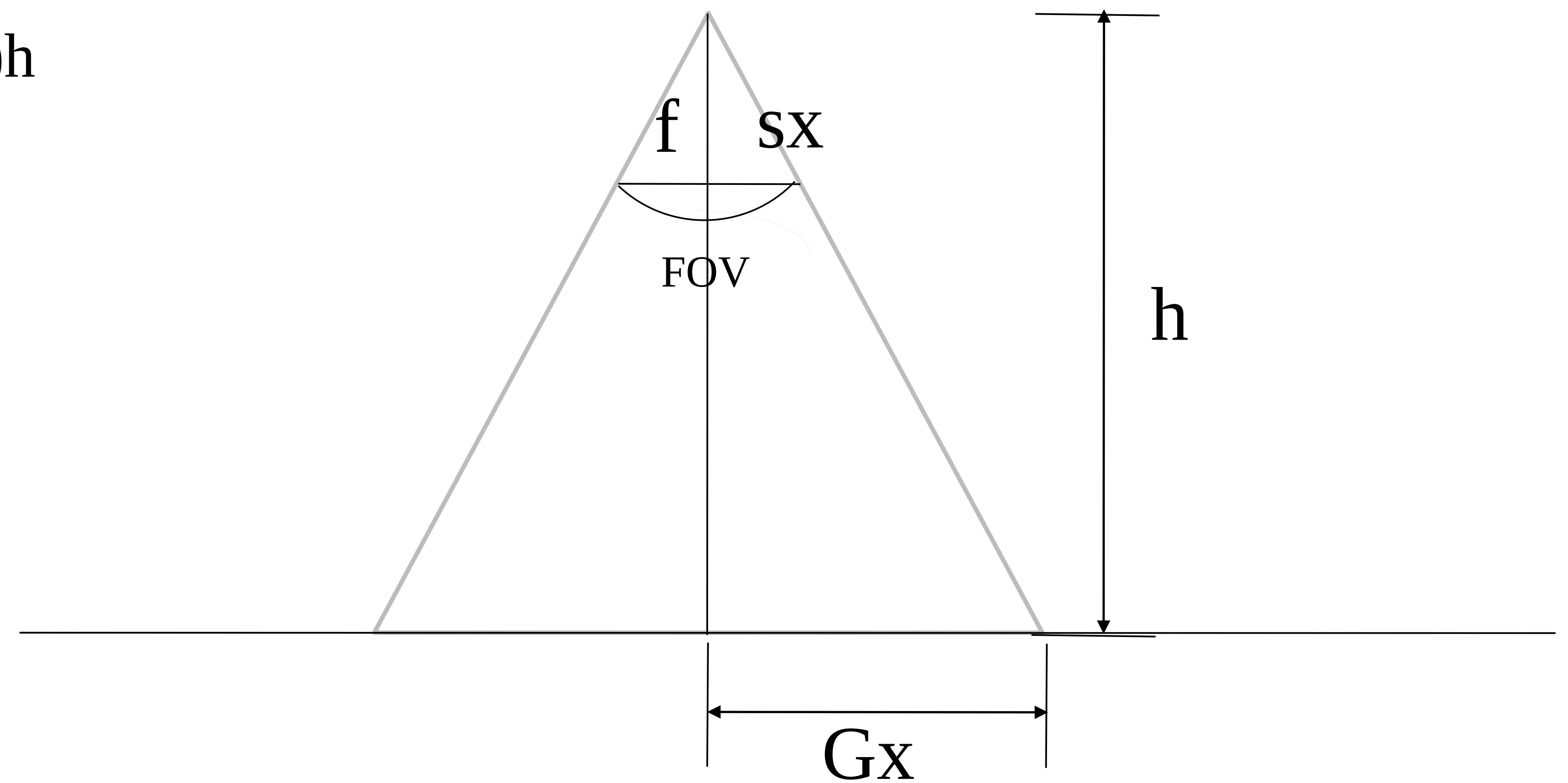
- Planar scene
- Camera plane parallel to ground plane
- Optical flow measures x,y shift in pixel
- Z and metric scale from altitude sensor



The optical flow idea

$$\frac{\mathrm{Gx}}{\mathrm{h}} = \frac{sx}{f} = \tan\left(\frac{\mathrm{FOV}}{2}\right)$$

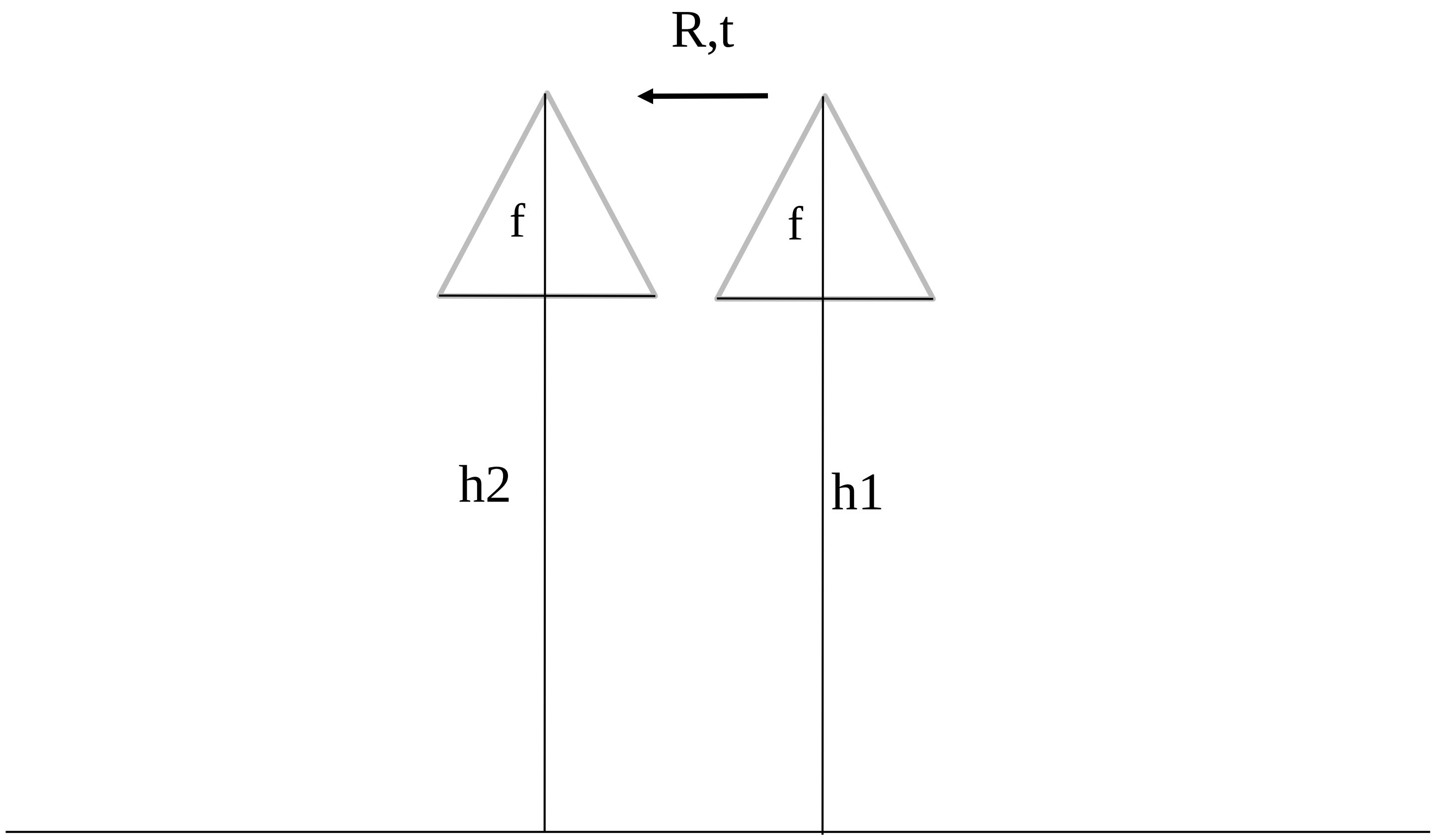
$$\mathrm{Gx} = \frac{sx}{f}h = \tan\left(\frac{\mathrm{FOV}}{2}\right)\mathrm{h}$$



640 pixel, but how many meters?

The optical flow idea

$$Gx = \frac{sx}{f} h$$

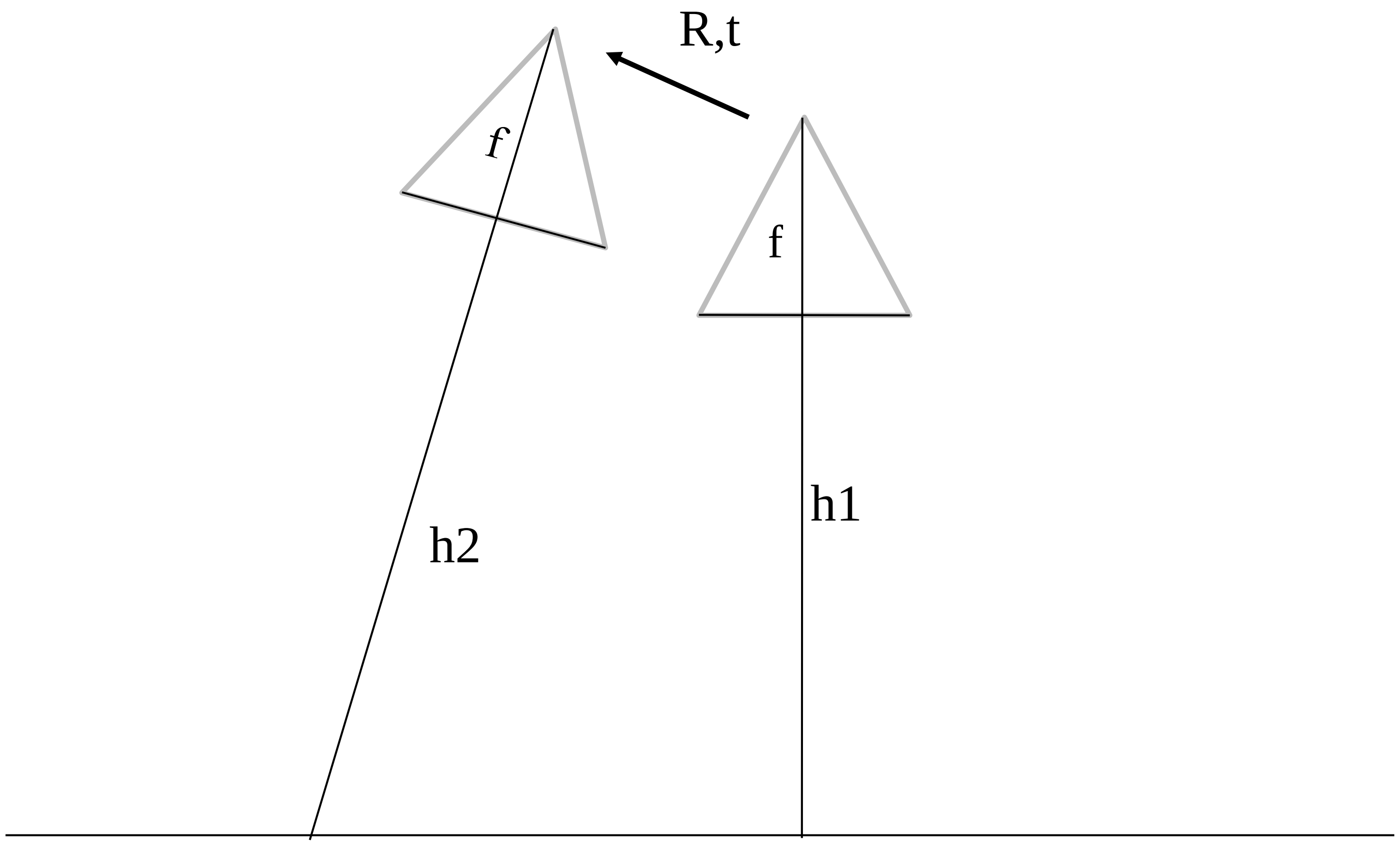


The optical flow idea

$$I2 = H * I1$$

$$p2 = H * p1$$

$$H = R + \frac{1}{d} T N^T$$

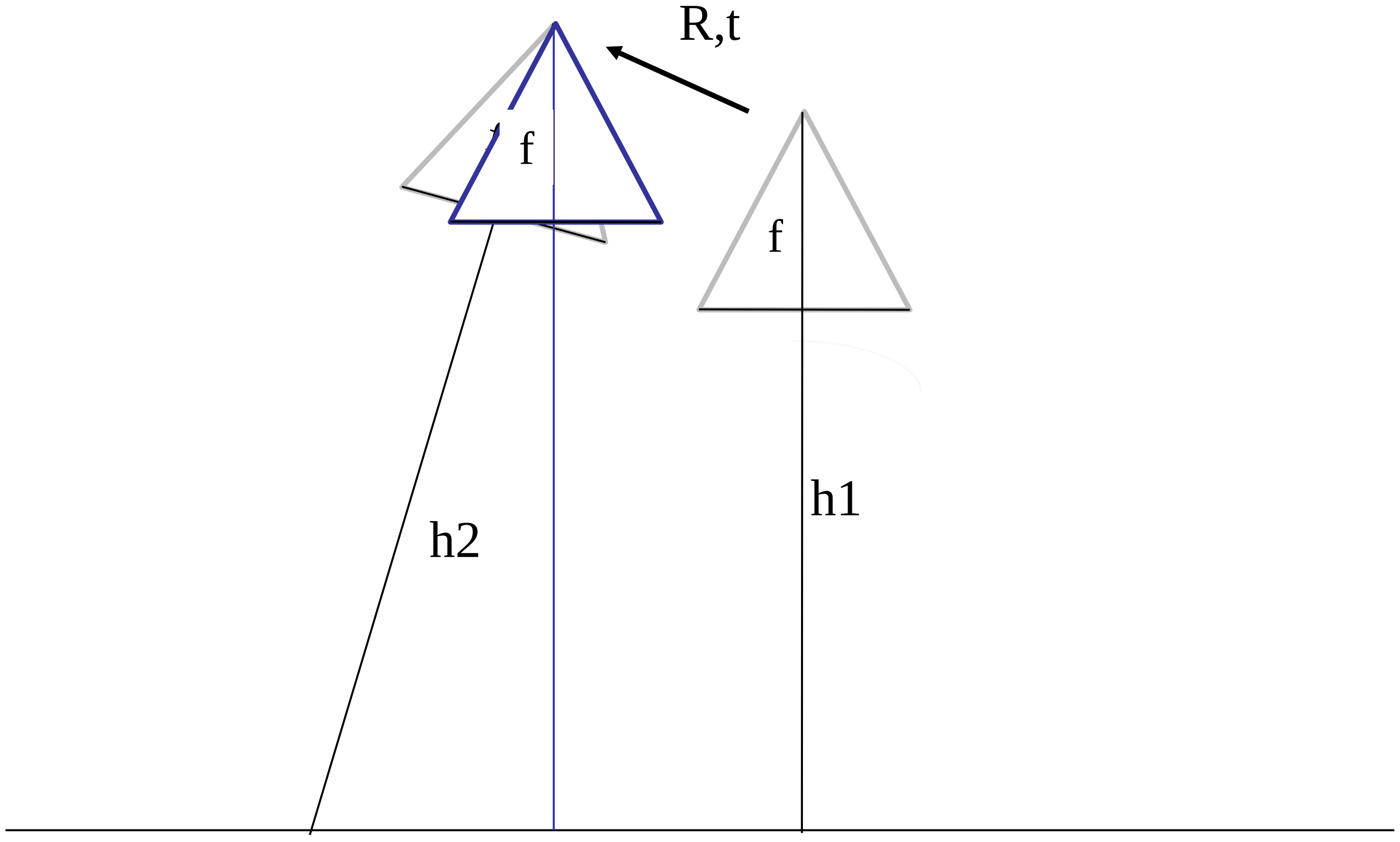


The optical flow idea

$$I2 = H * I1$$

$$p2 = H * p1$$

$$H = R + \frac{1}{d} T N^T$$

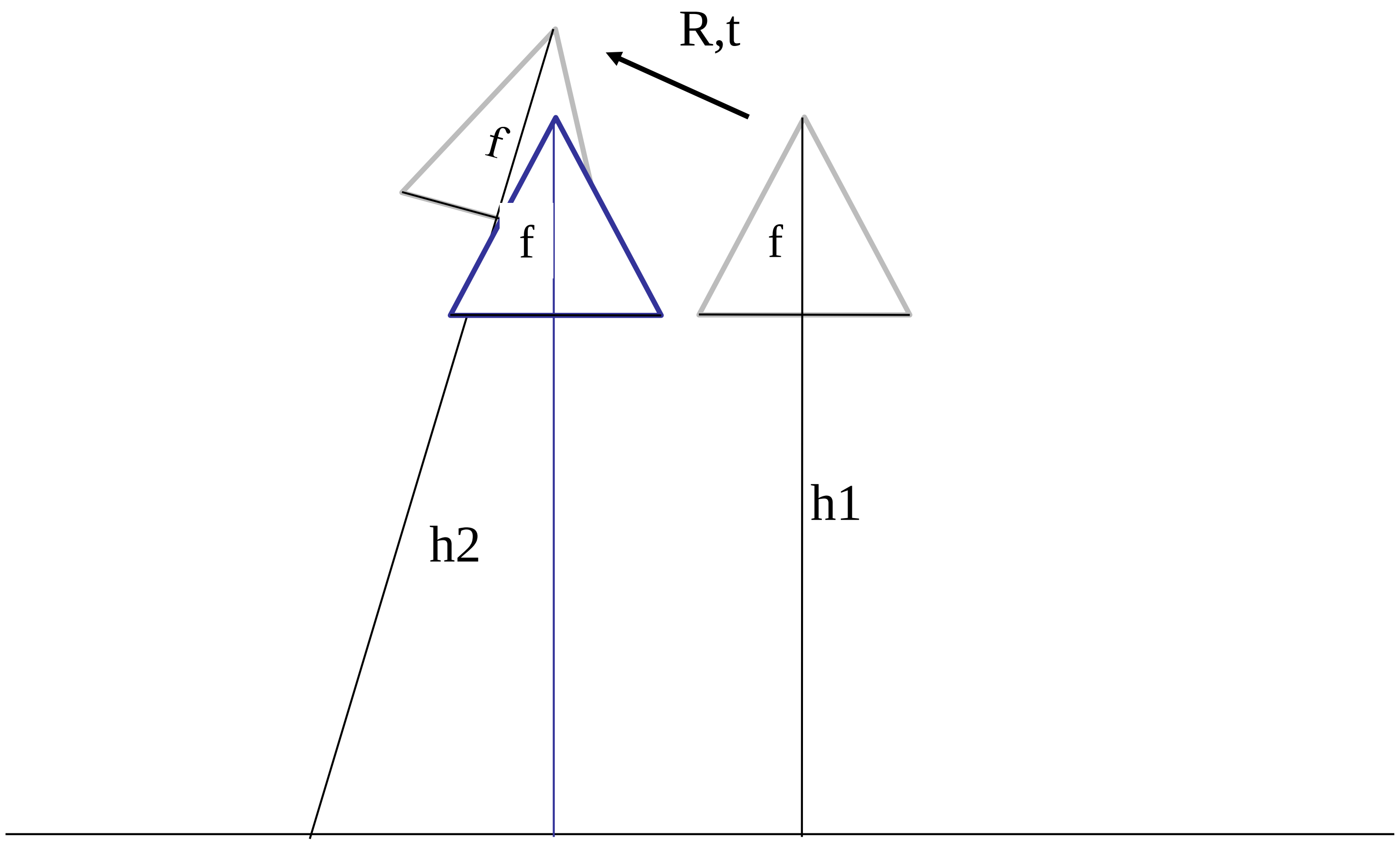


The optical flow idea

$$I2 = H * I1$$

$$p2 = H * p1$$

$$H = R + \frac{1}{d} T N^T$$



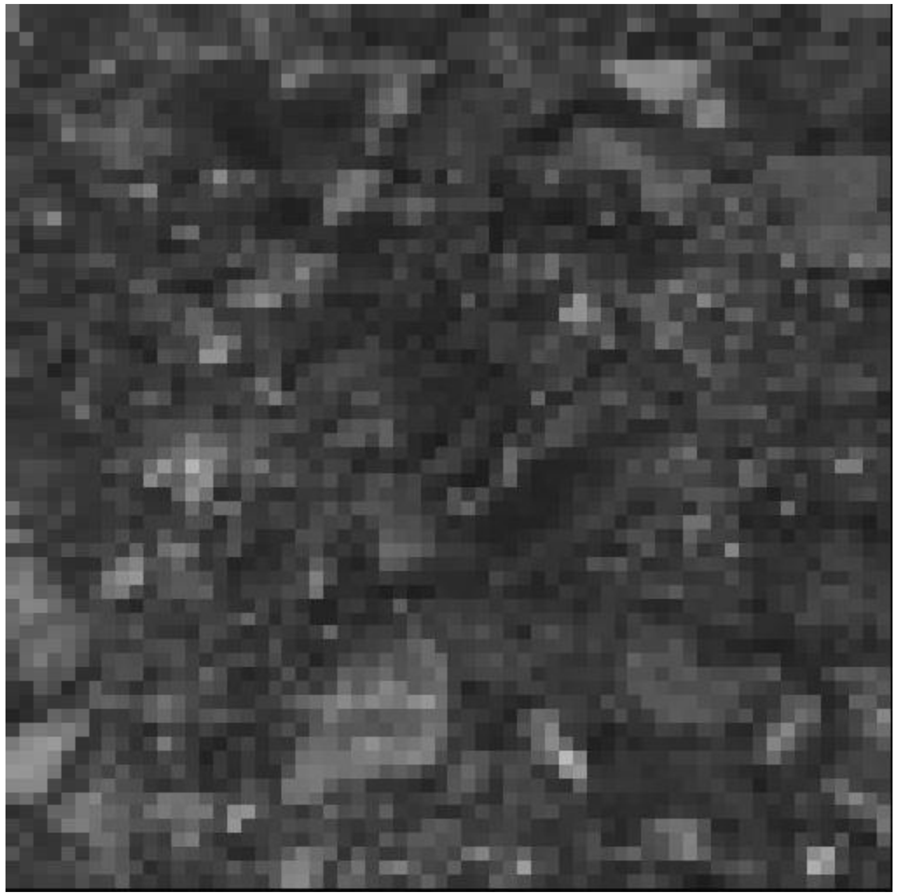
Optical flow camera -PX4Flow

- Smart camera module
  - 752Hx480V (60fps), 188Hx120V (250fps), 16mm lens
  - ARM Cortex M4 (168 MHz, 192 KB RAM, single precision floating point operations)
  - MEMS gyroscope (L3GD20)
  - Ultrasound sensor

- Outputs speed
- Serial interface (Mavlink)
- ROS node (http://www.ros.org/wiki/px4flow_node)

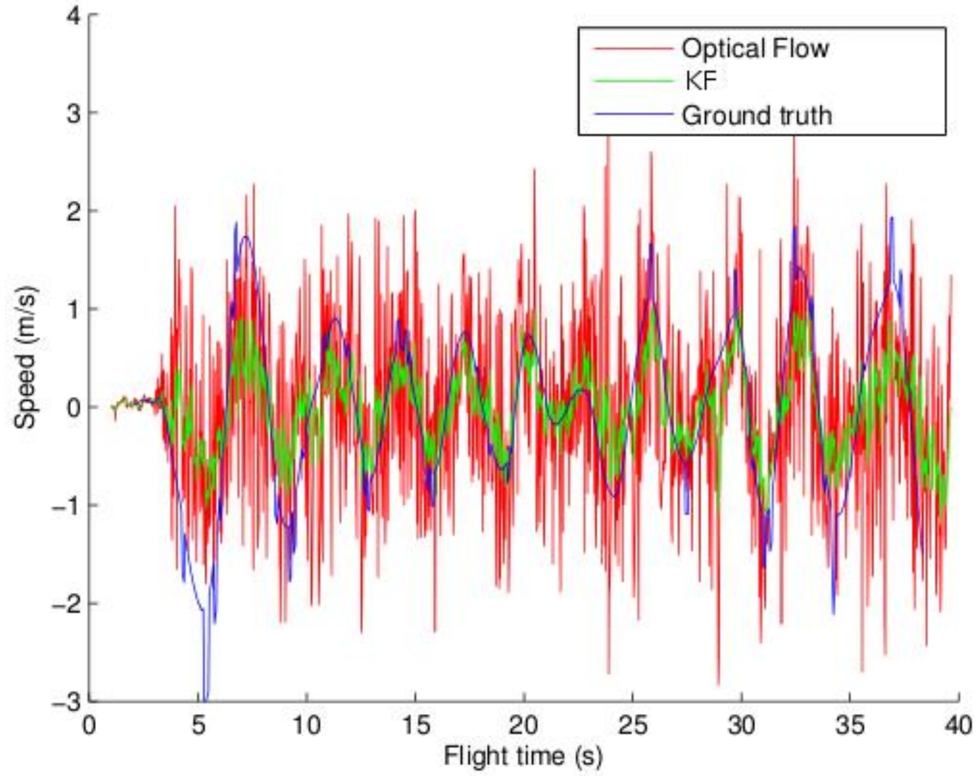
Optical flow camera -PX4Flow

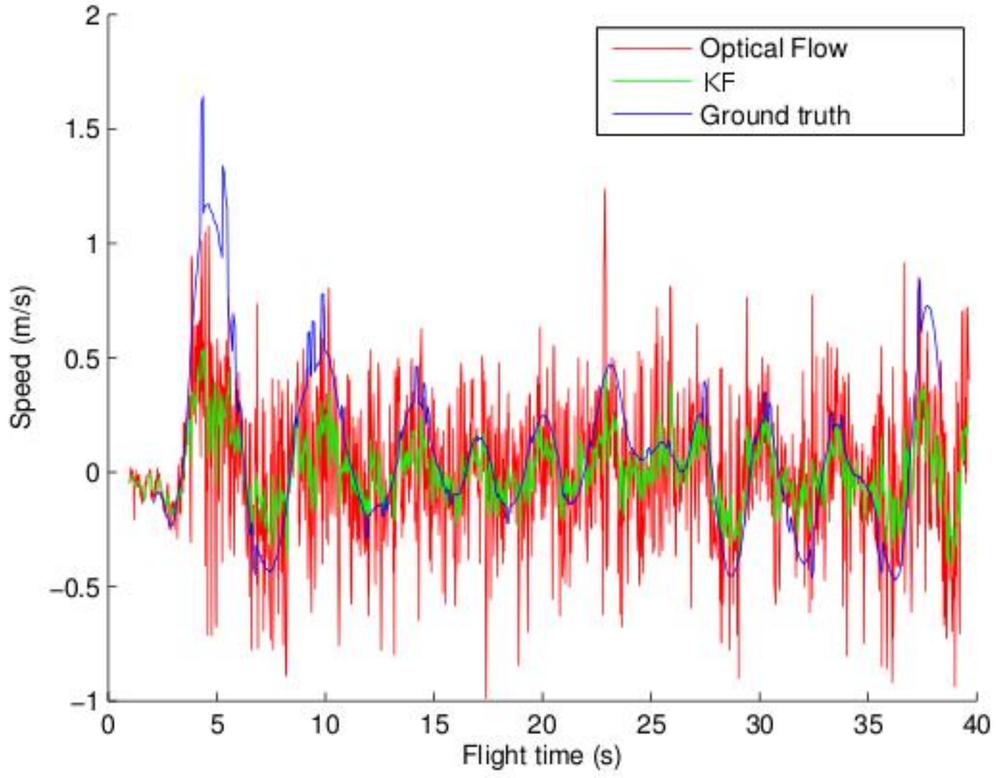
- 64x64 pixel, 250Hz
- SAD optical flow computation
- 8x8 pixel blocks within a 4x4 pixel search range
- Histogram voting
- Subpixel refinement
- Removal of orientation component
- Outputs speed



street texture as seen from the flow sensor from 0.8 m altitude through a 16 mm M12 lens.

Optical flow camera -PX4Flow

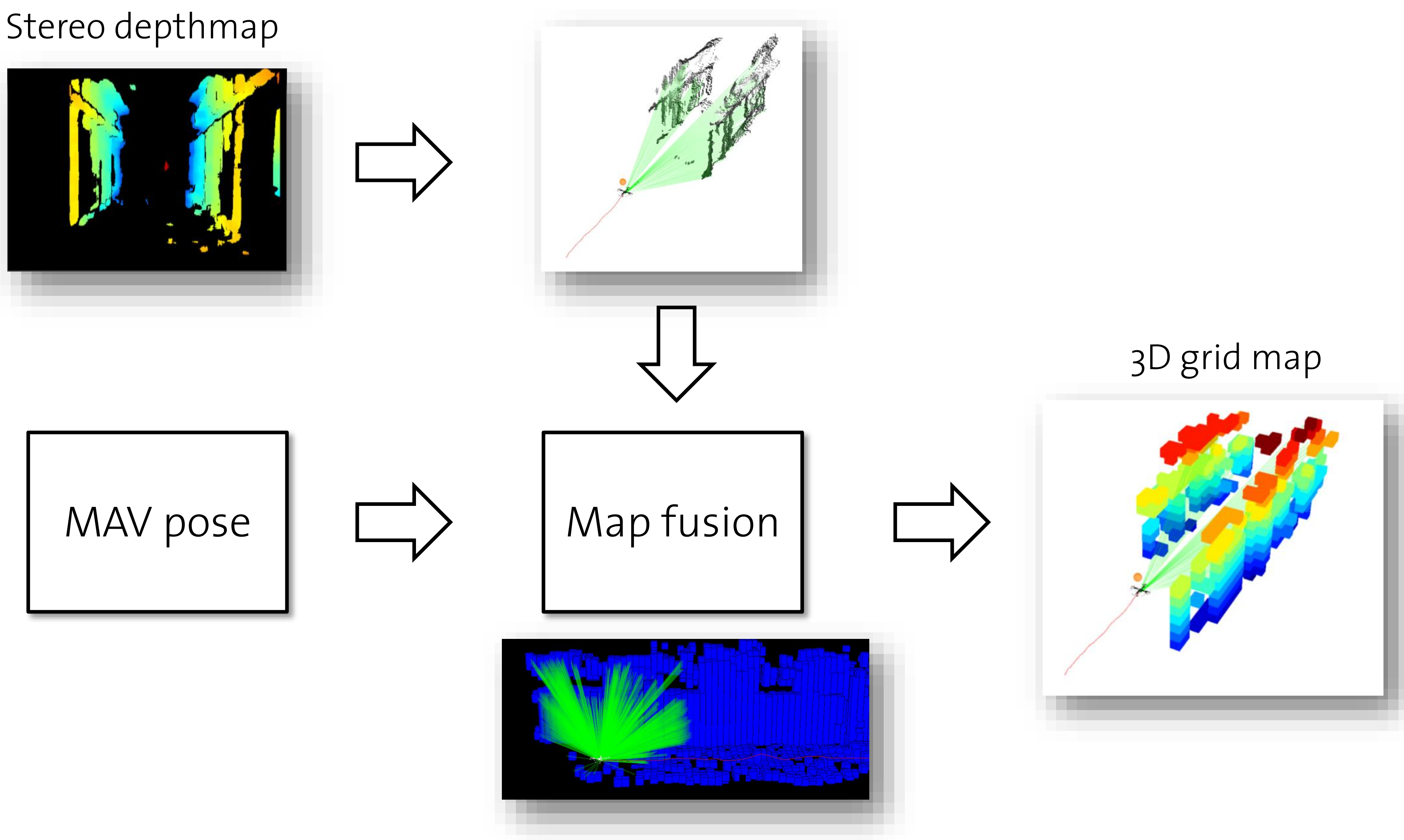




- OF speed filtering necessary (Kalman filter)

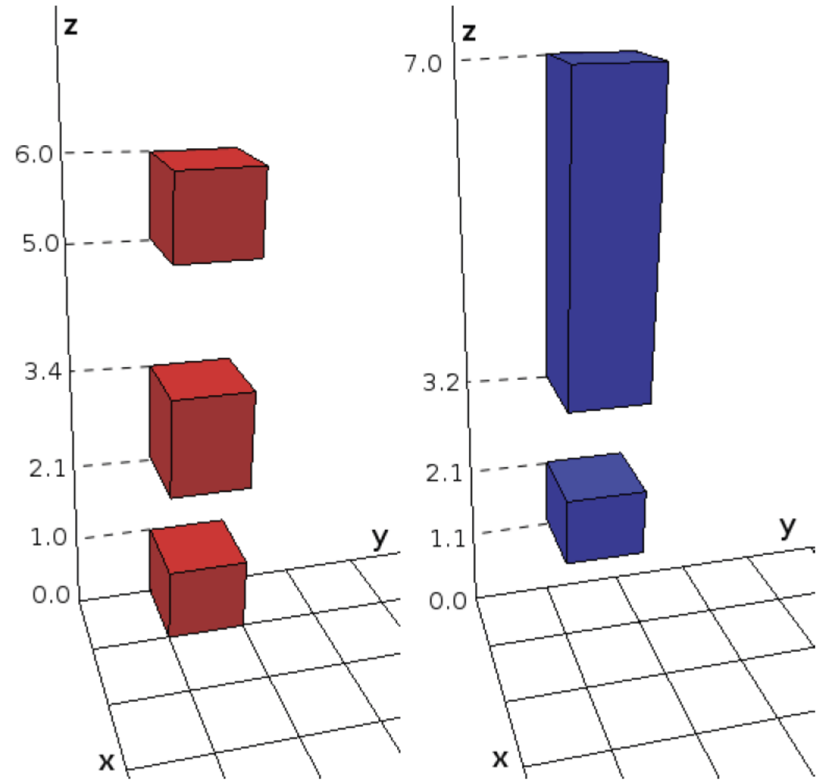
Honegger et al. An Open Source and Open Hardware Embedded Metric Optical Flow CMOS Camera for Indoor and Outdoor Applications, ICRA 2013

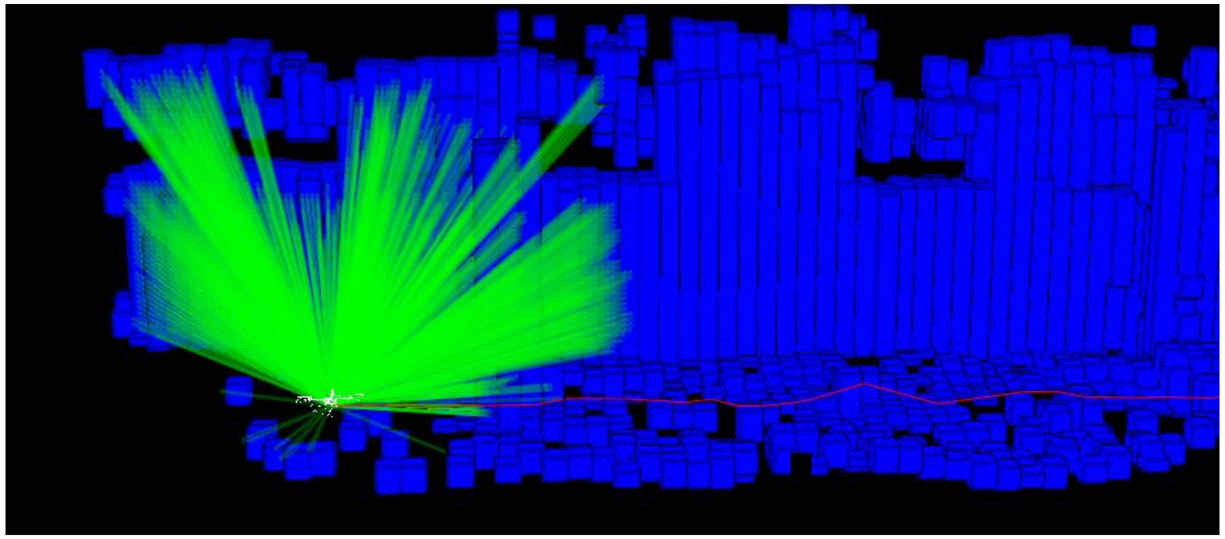
3D Mapping



Multi-volume occupancy grid

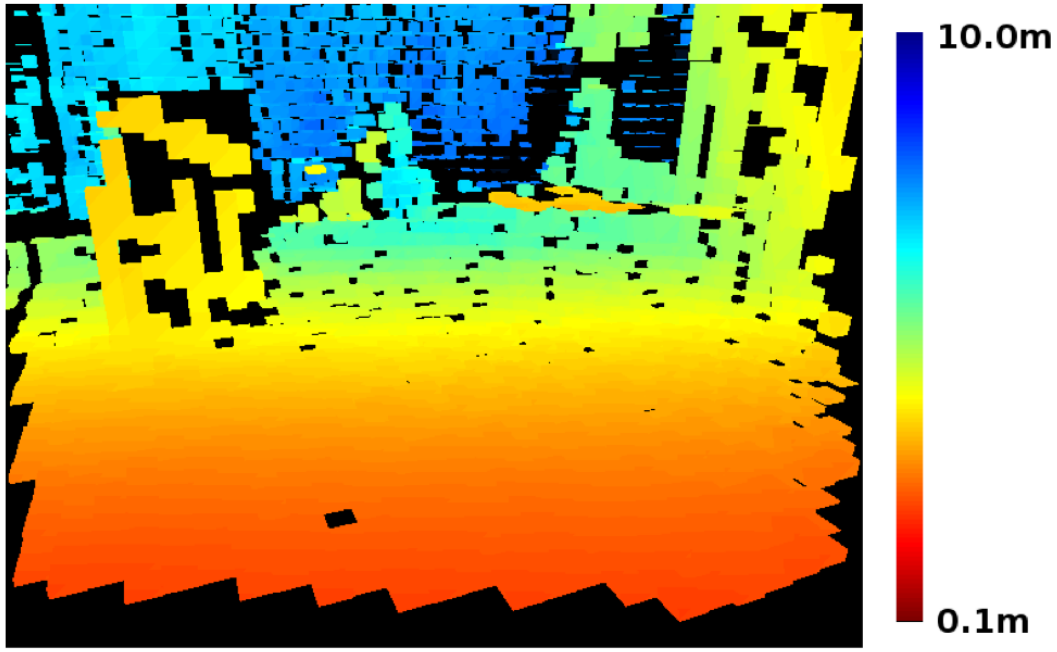
- Multi-volume occupancy grid implementation [Morris 2010].
  - Group sensor readings into continuous vertical volumes which are stored in a 2D grid.
  - Record both positive and negative readings.
    - Models free and occupied space.

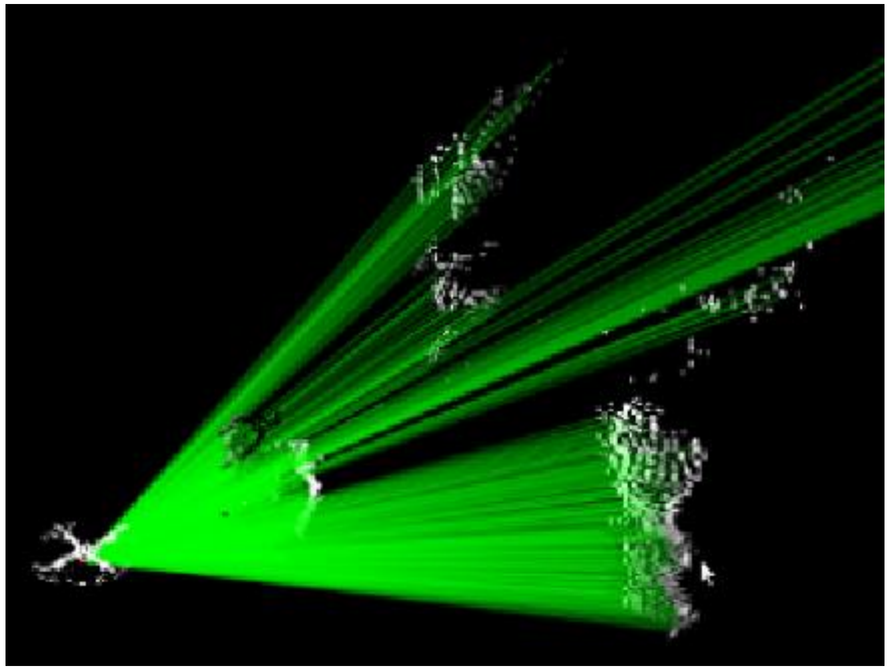




Multi-volume occupancy grid

- Update 3D grid with distance measurements
- Downsample range data from stereo / Kinect to a virtual scan.
  - Outlier removal and efficient occupancy grid updates.
- Each ray in a virtual scan measures the median distance to the range points falling in an angular interval.

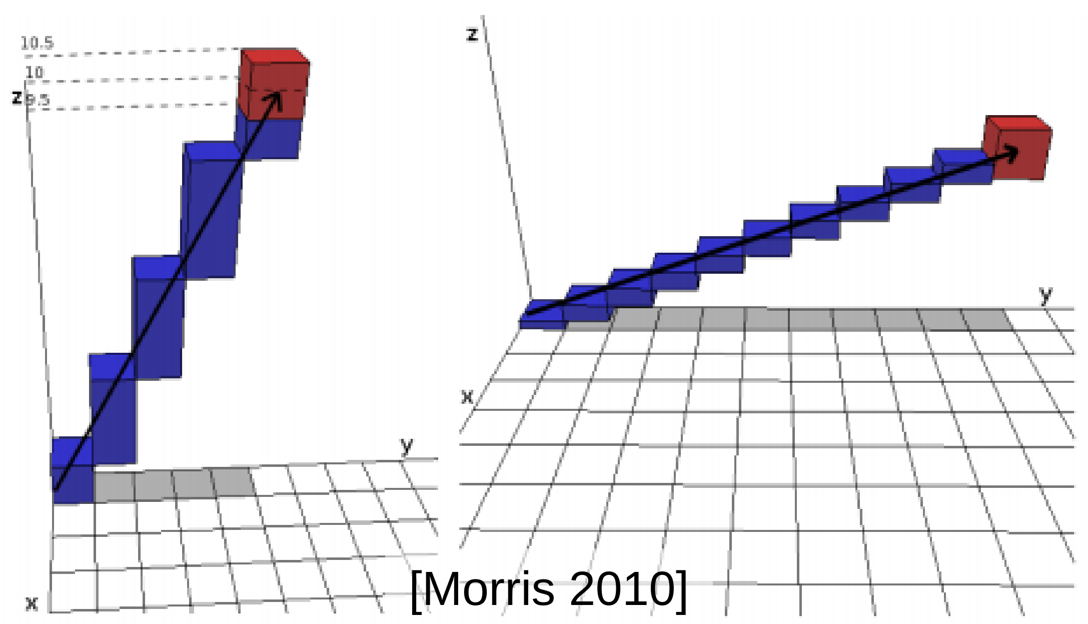




Multi-volume occupancy grid

Updating the map:

- For each ray in the virtual scan,
  - Traverse in order the cells intersected by the ray.
  - Insert negative volumes in the cells until we reach the endpoint of the ray at which we insert a positive volume.
  - Merge overlapping volumes (changes densities of volumes)



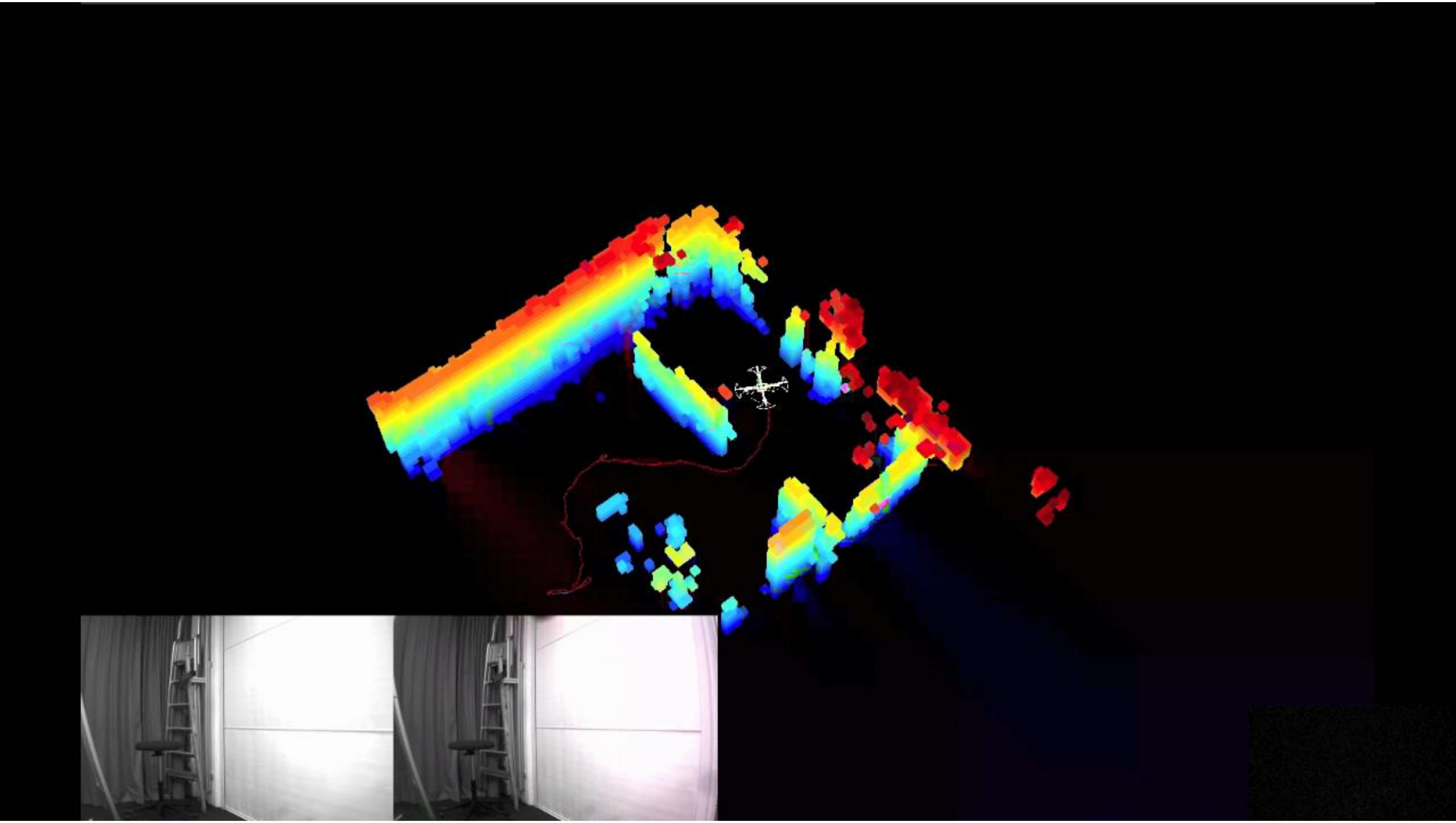
[Morris 2010]

Multi-volume occupancy grid

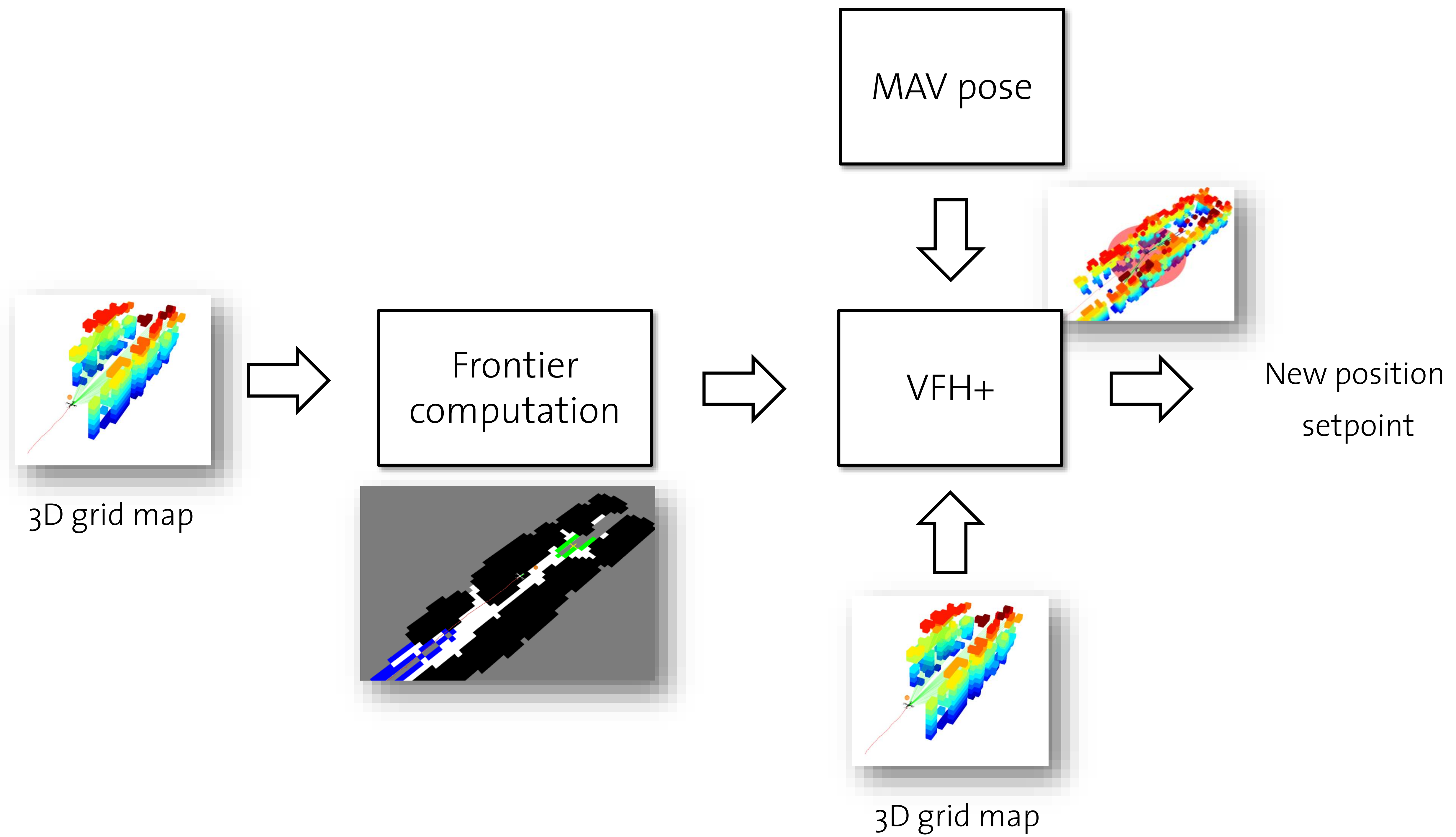
- Occupancy probabilities can be computed for any specific coordinate x,y,z
- Typically one can extract occupancy probabilities for different height planes

$$\text{Occupancy probability of } p = \frac{\text{Occupancy density of positive volume } p \text{ is in}}{\text{Occupancy density of positive \& negative volumes } p \text{ is in}}$$

Video – Online Mapping Test



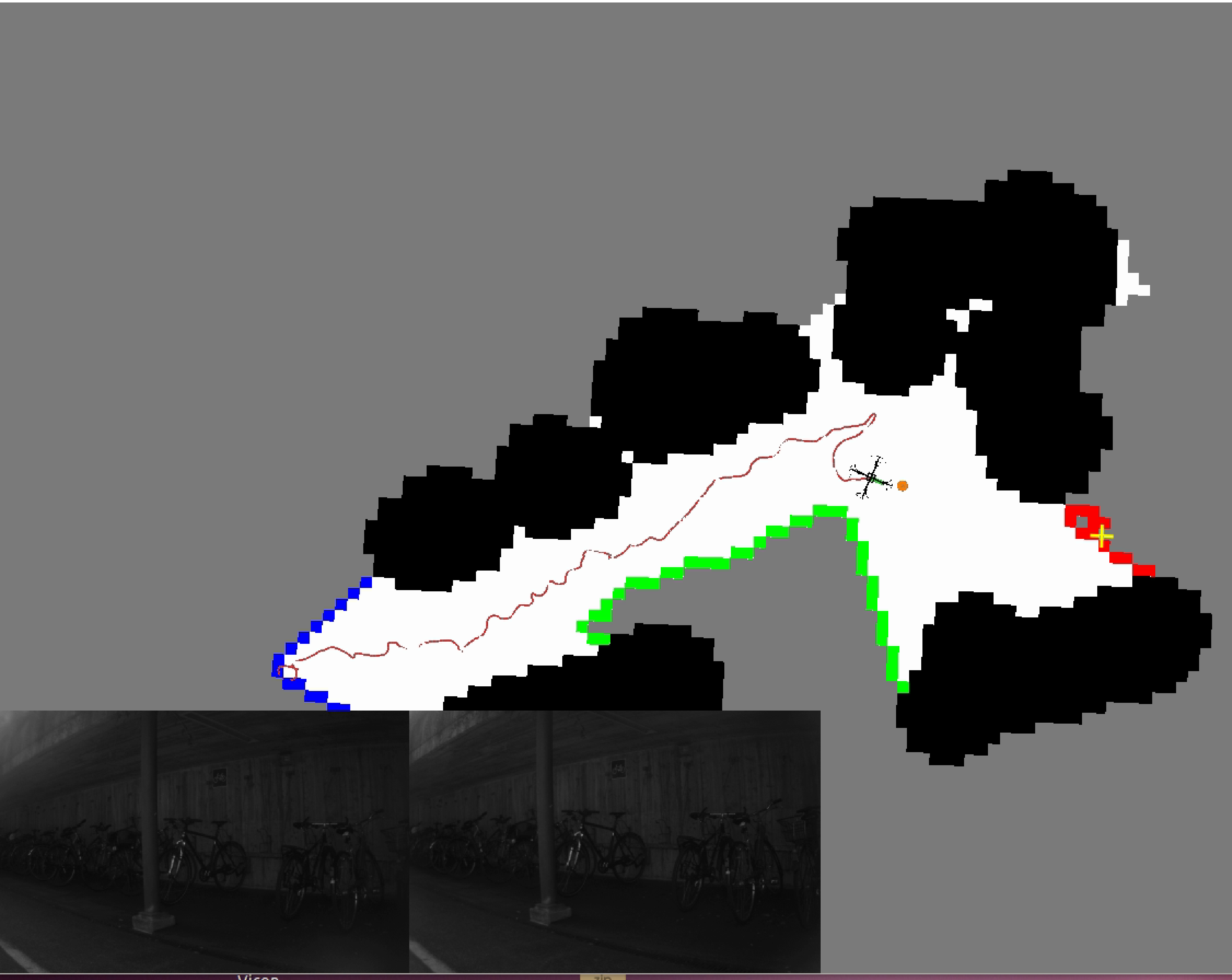
Exploration



Frontier based exploration

- Method suitable for exploration in grid map (occupancy grid) environment representation
- Frontiers are boundaries between known (sensed) and unknown (unsensed) area.
- MAV is directed to centroid of frontiers

Frontier based exploration



Black ... Occupied Cells

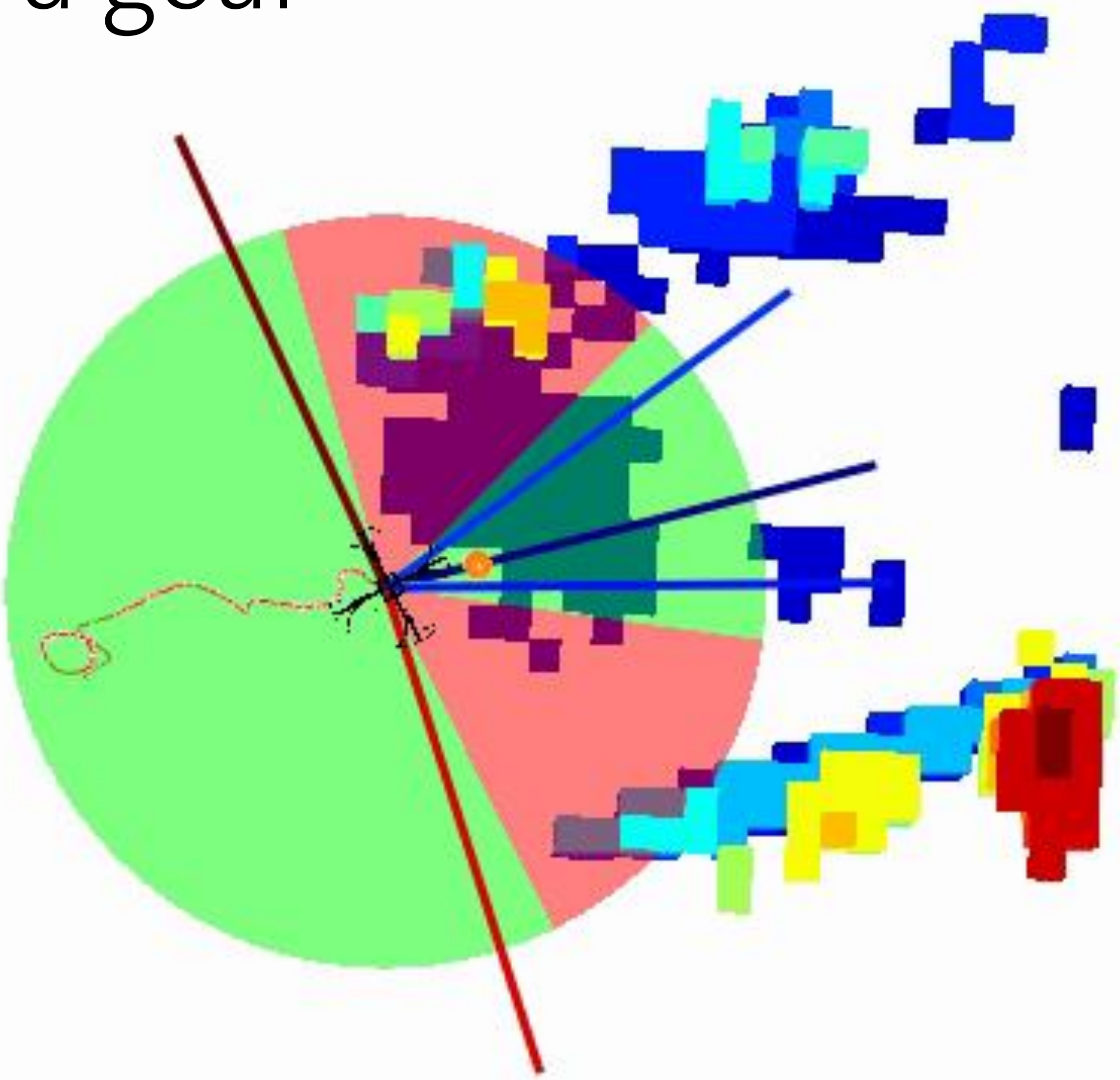
White ...Free Cells

Grey ... Unknown Cells

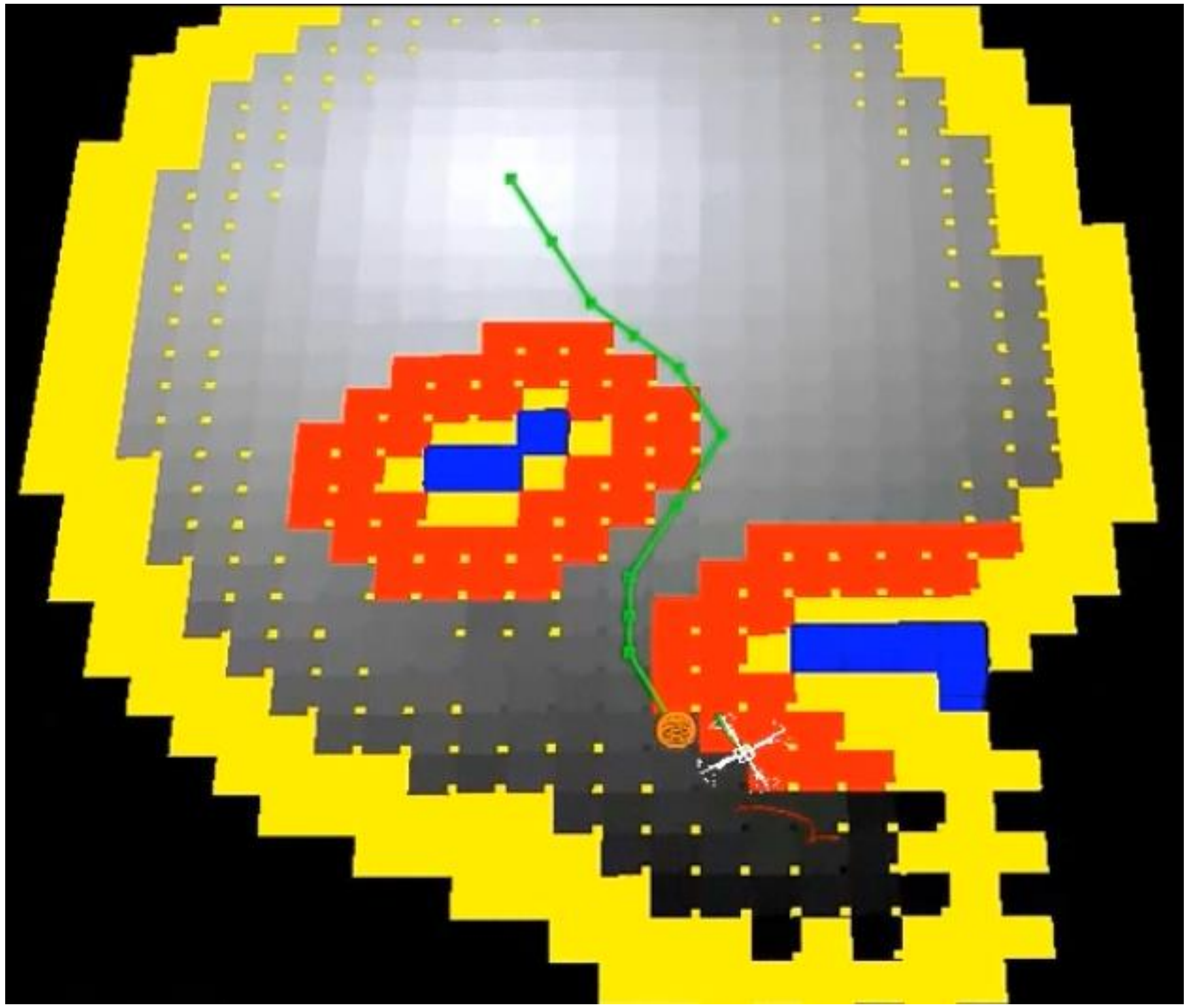
Colored ... Frontier Cells

Vector Field Histogram (VFH)

- Only 1 waypoint at a time
- Fly discrete steps toward goal



State lattice



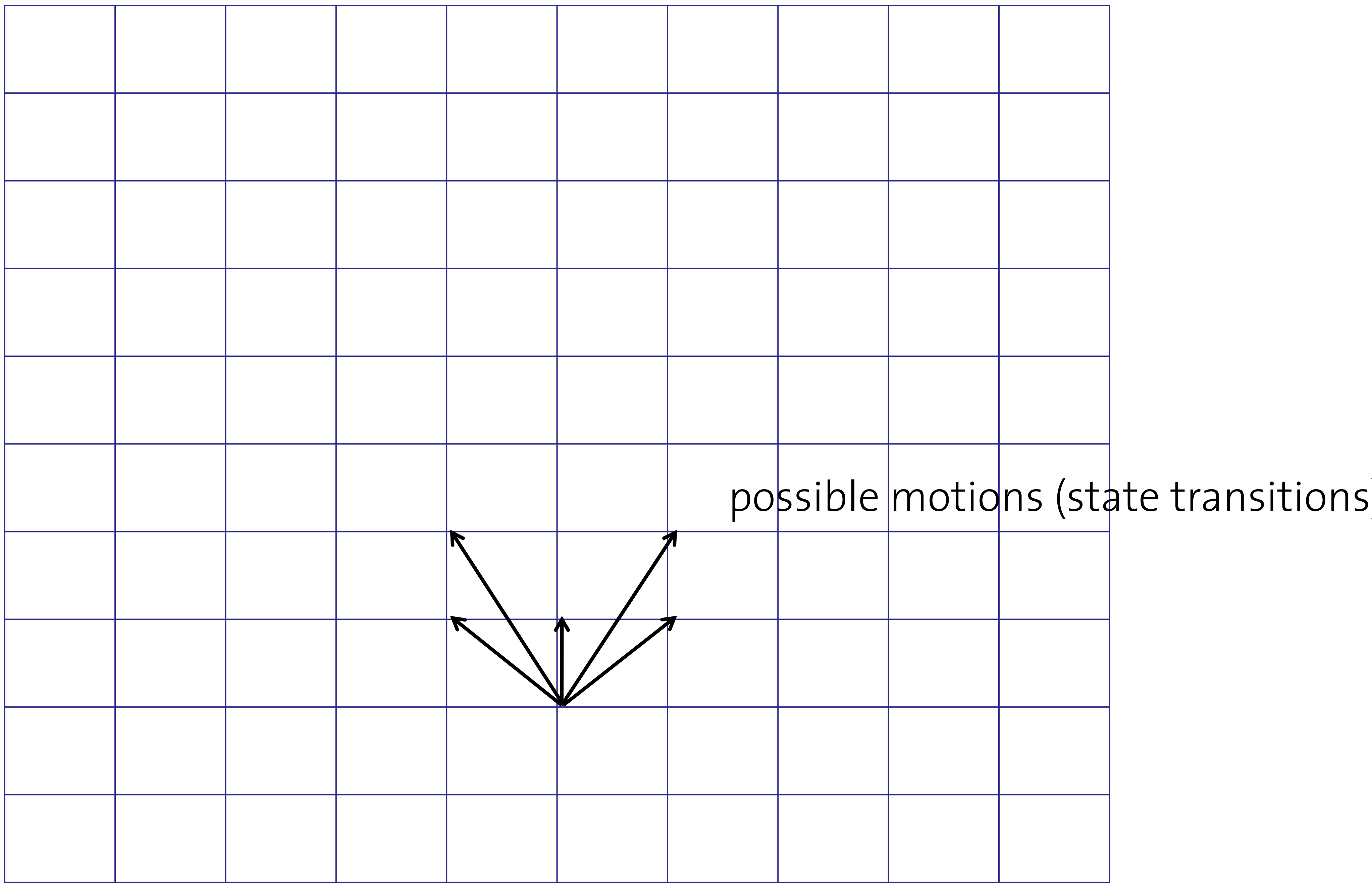
Blue Cell	Occupied cell in 3D occupancy grid map
Red Cell	Non-traversable cell
Yellow Cell	Unexplored cell
Grayscale Cell	Traversable cell (The more white, the nearer to the goal)
Green Line	Planned flight path
Red Line	Trajectory history
Orange Sphere	Current waypoint to follow

Safety Clearance = 0.75m

State lattice

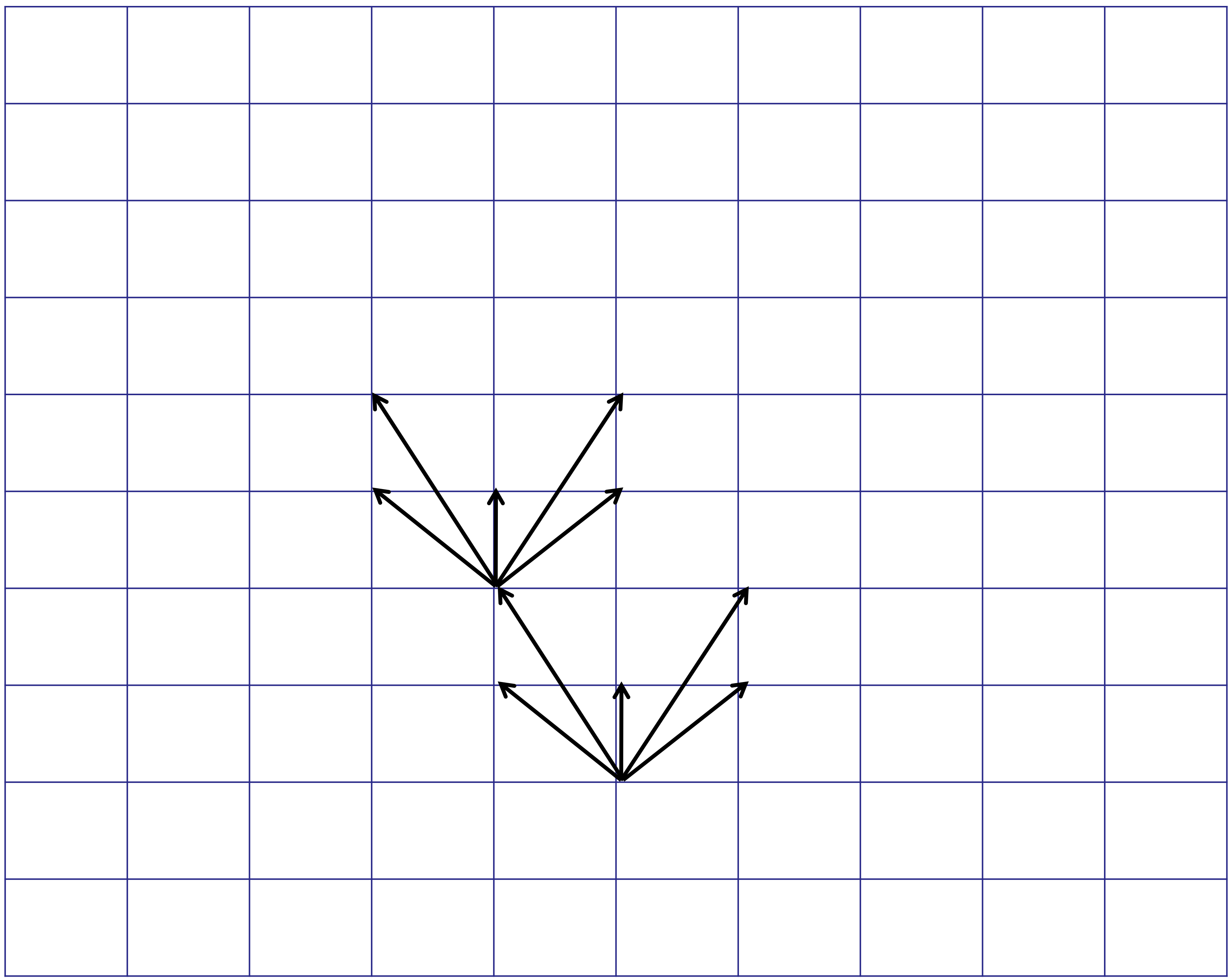
- State lattice concept [Pivtoraiko et al., 2009] can include mobility constraints
- For the MAV we only want to allow forward motion, because of forward looking cameras
- A state lattice is a discretization of the state space and continuous motion primitives that connect these states with edges (graph structure)
- The MAV flies at a fixed attitude; each node represents a 3D state $[x, y, \theta]$.

State lattice

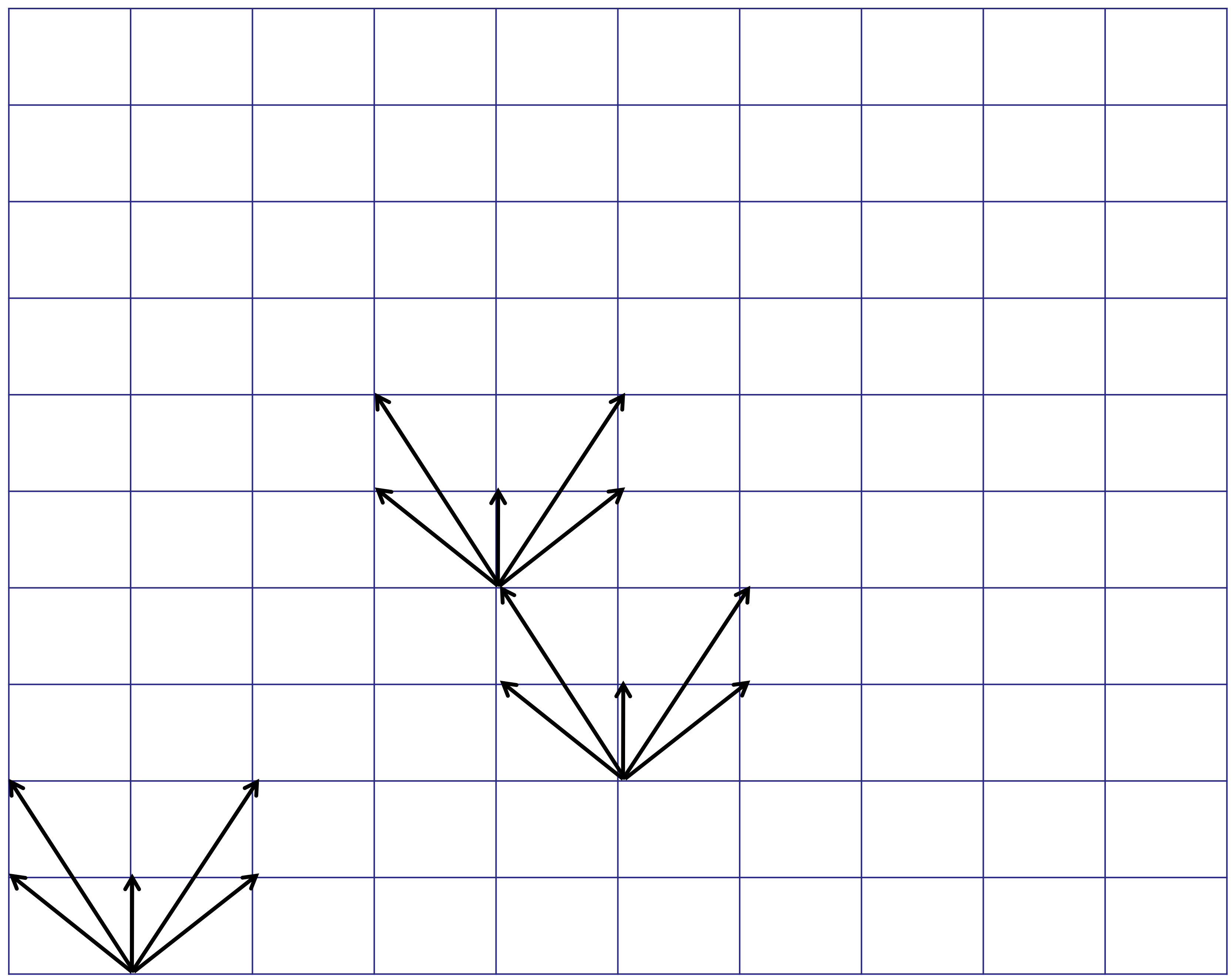


grid of all possible discrete states (here x,y)

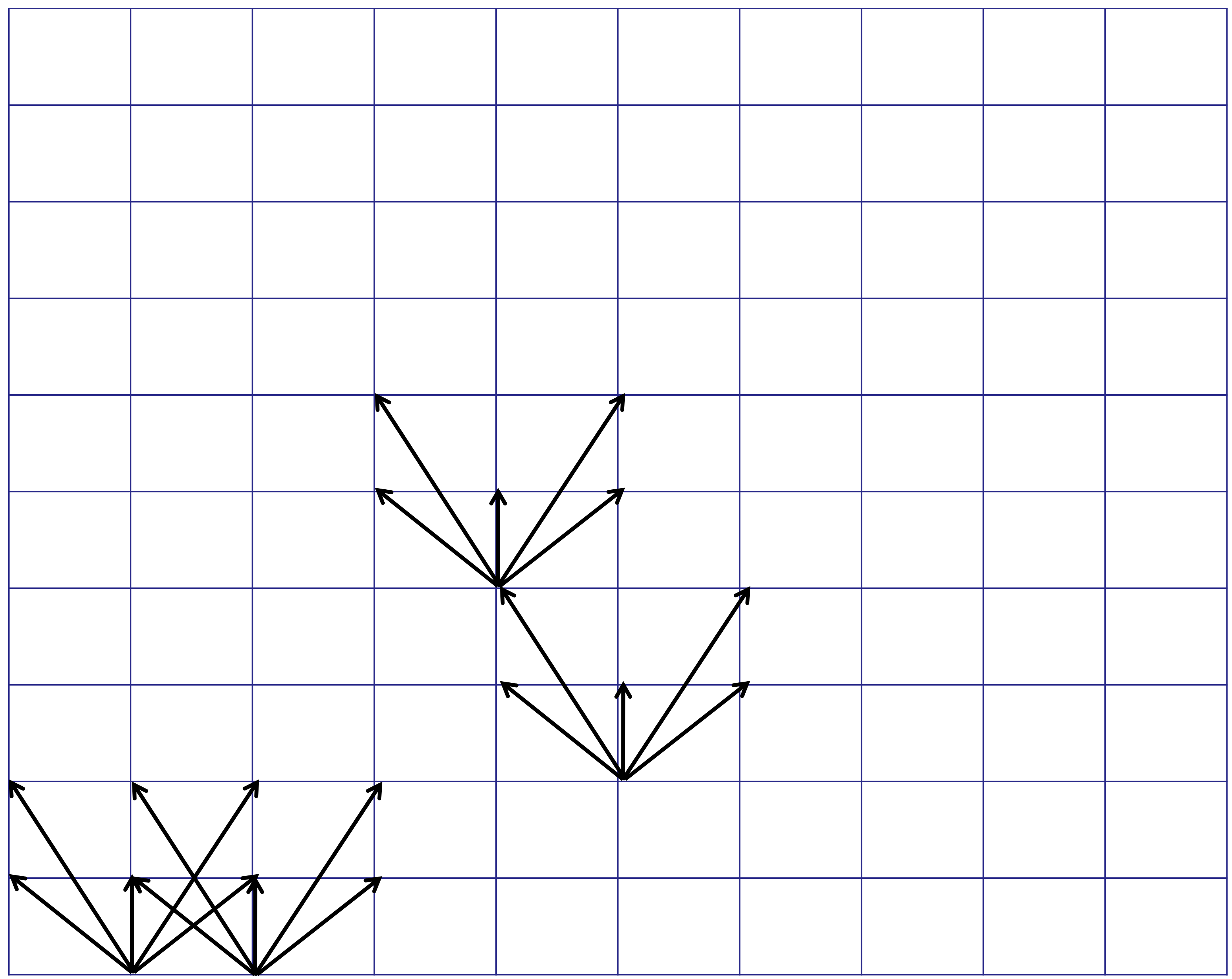
State lattice



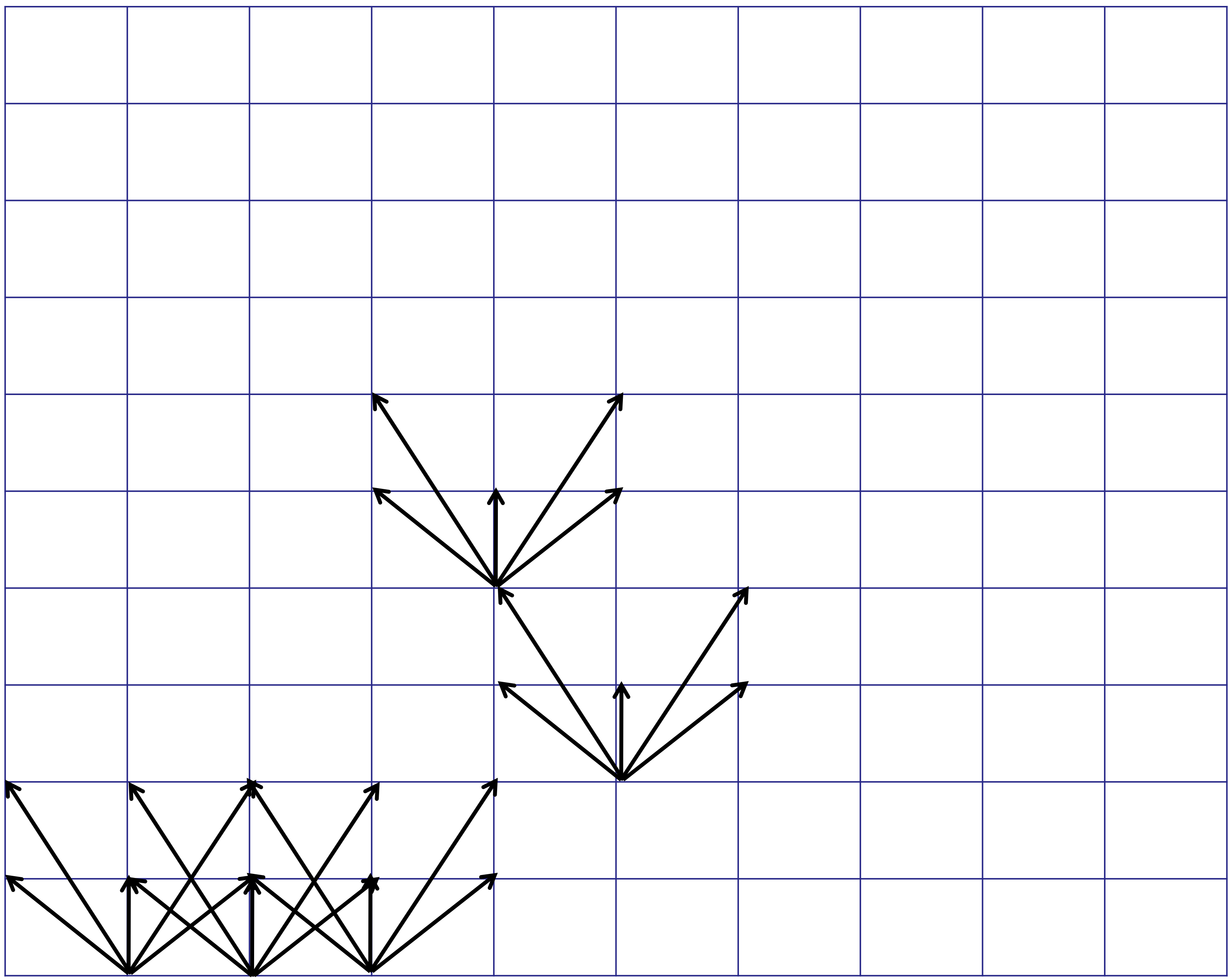
State lattice



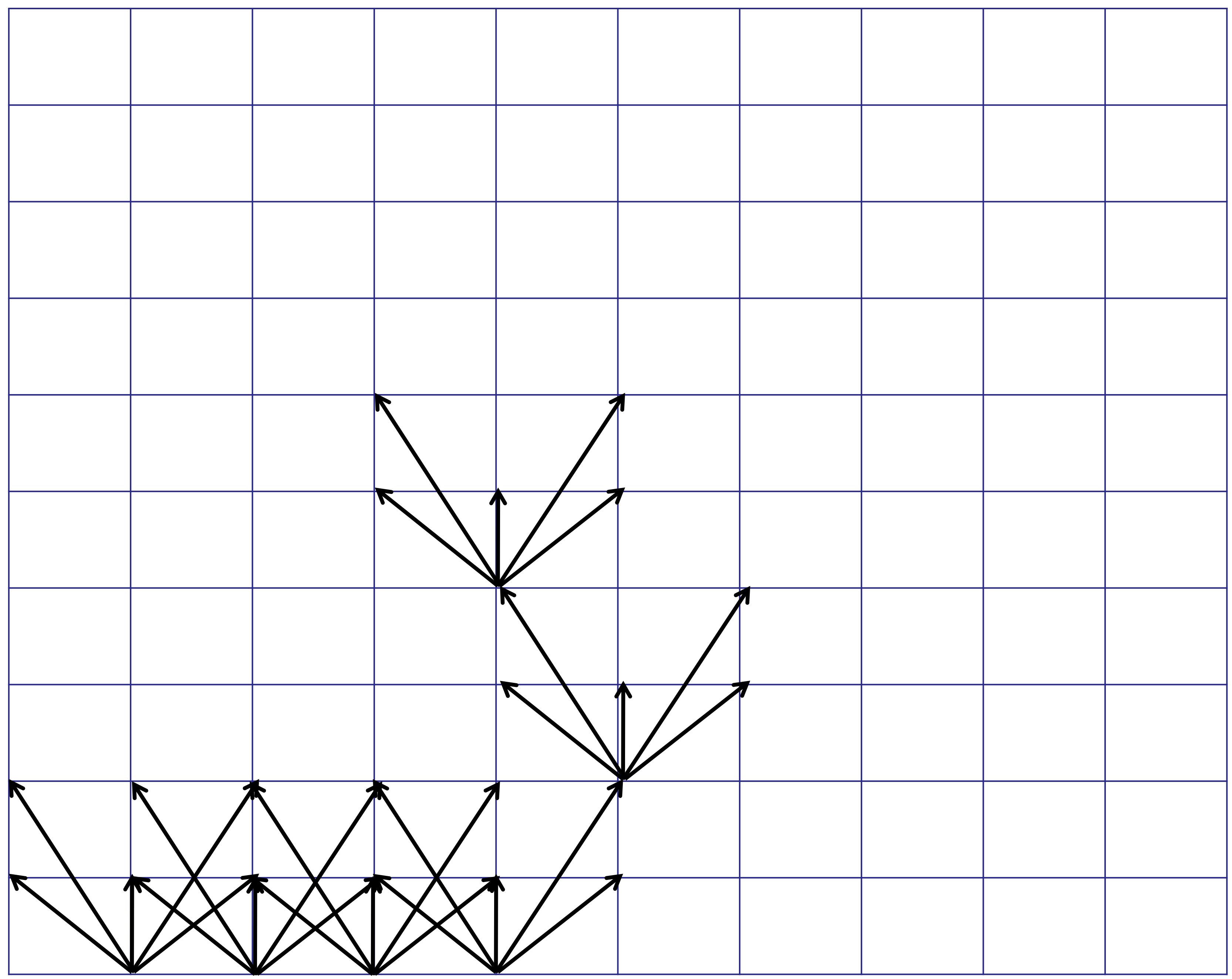
State lattice



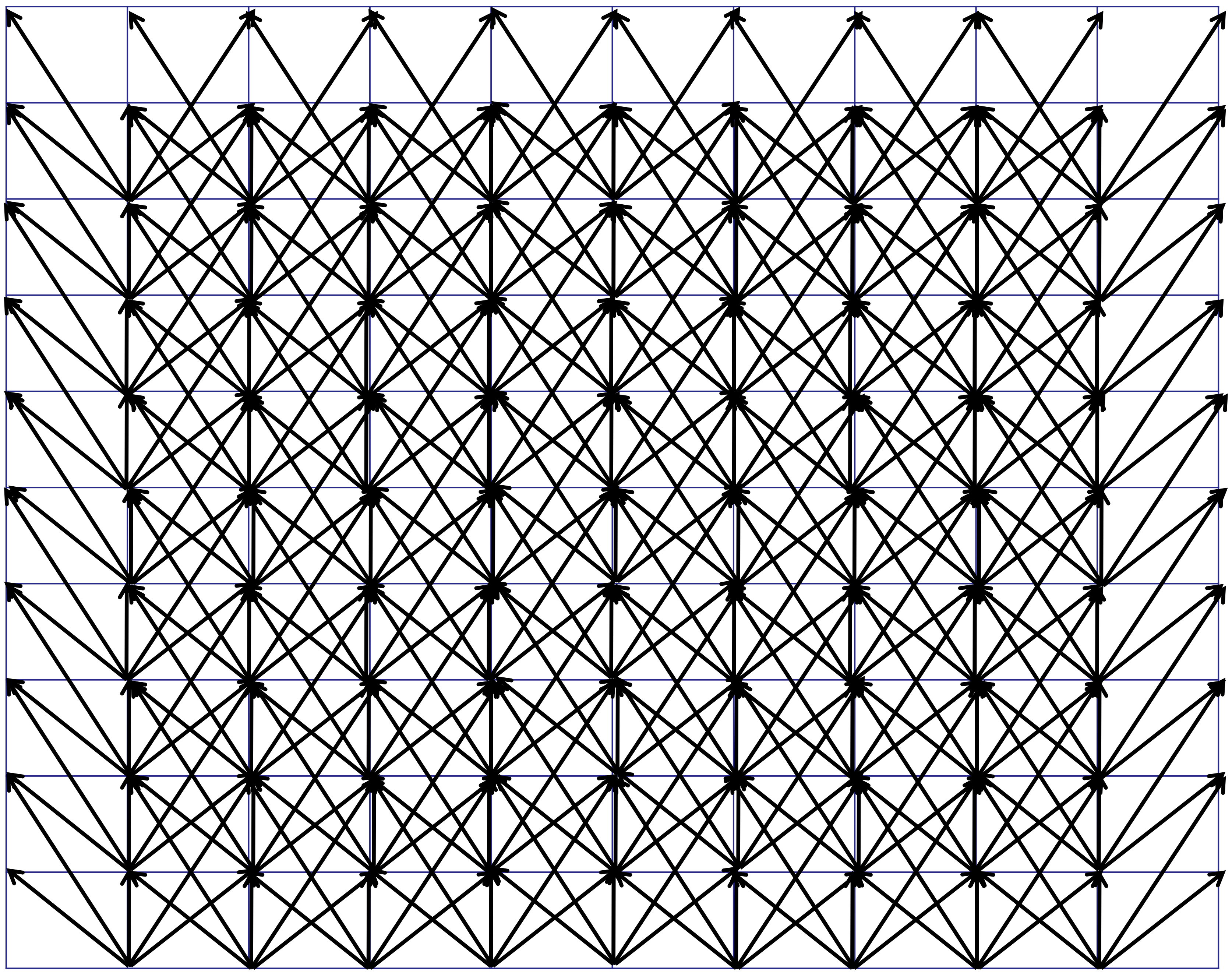
State lattice



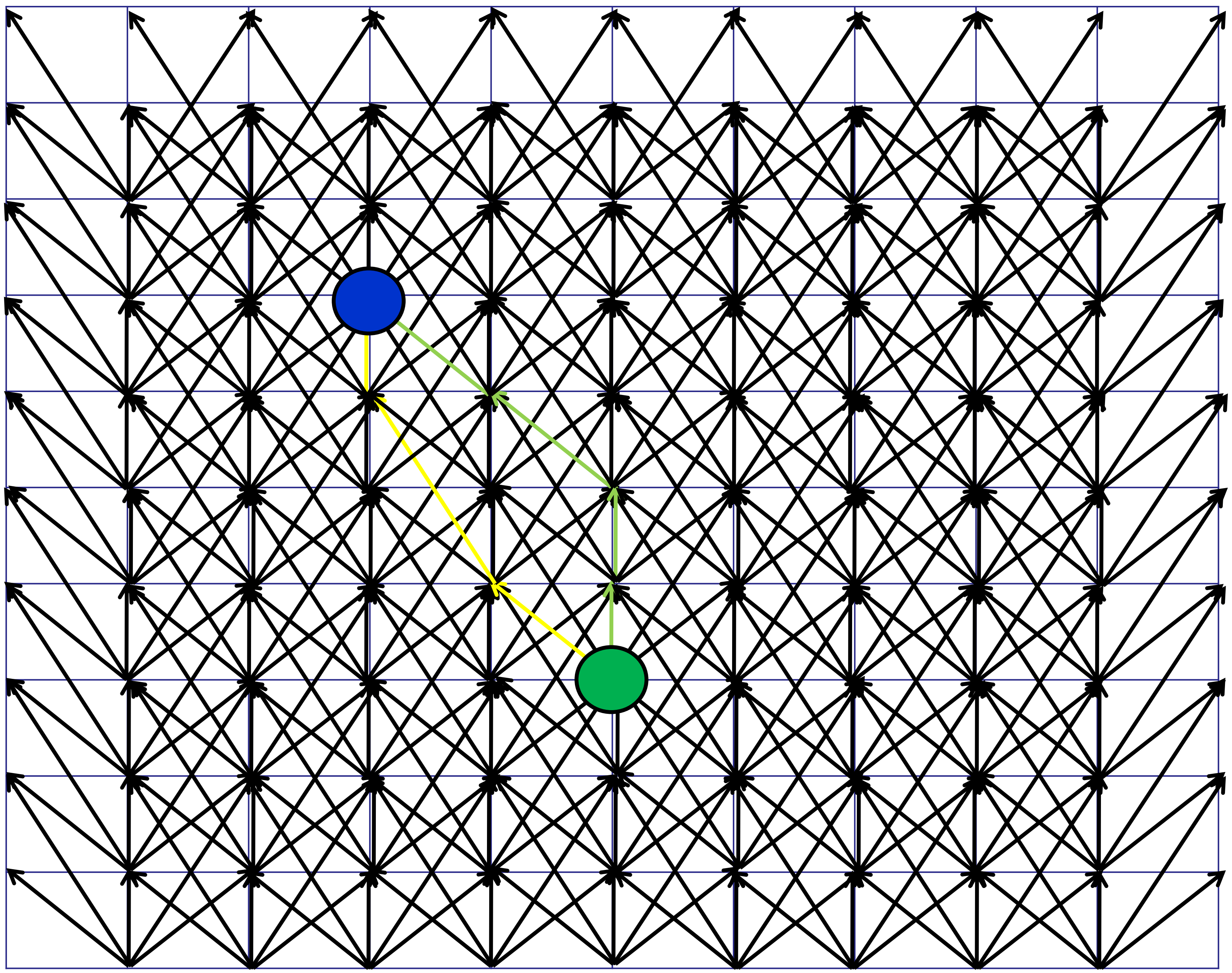
State lattice



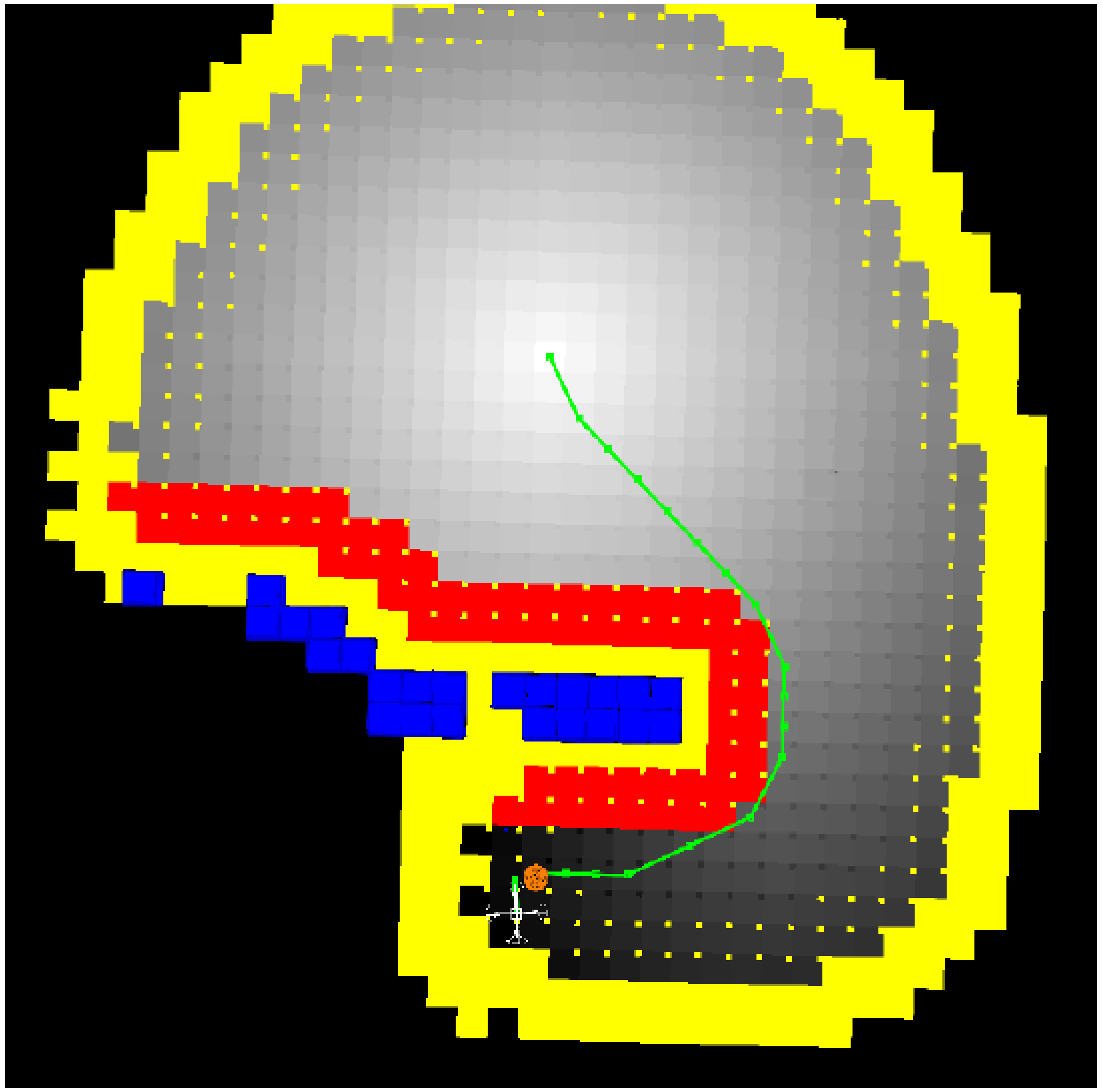
State lattice



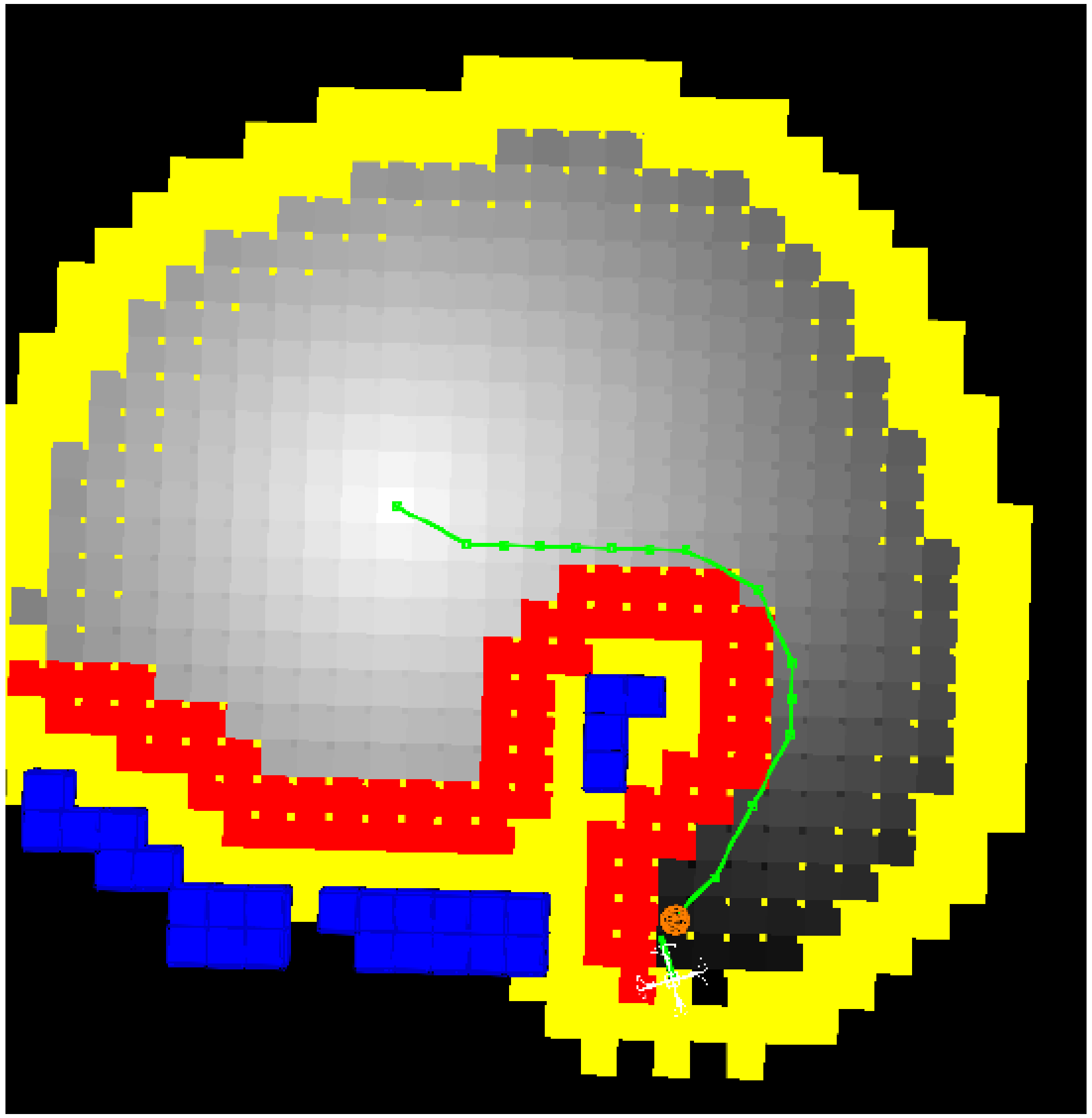
State lattice



Graph search



initial plan



new obstacle detected,
new plan computed

Obstacle detection and avoidance

[ICRA2011]

Lessons learned

- System components for autonomous MAV‘s
- Egomotion estimaton using optical flow
- 3D environment mapping using multi-volume occupancy grids
- Frontier based exploration
- Local navigation with VFH and state lattice