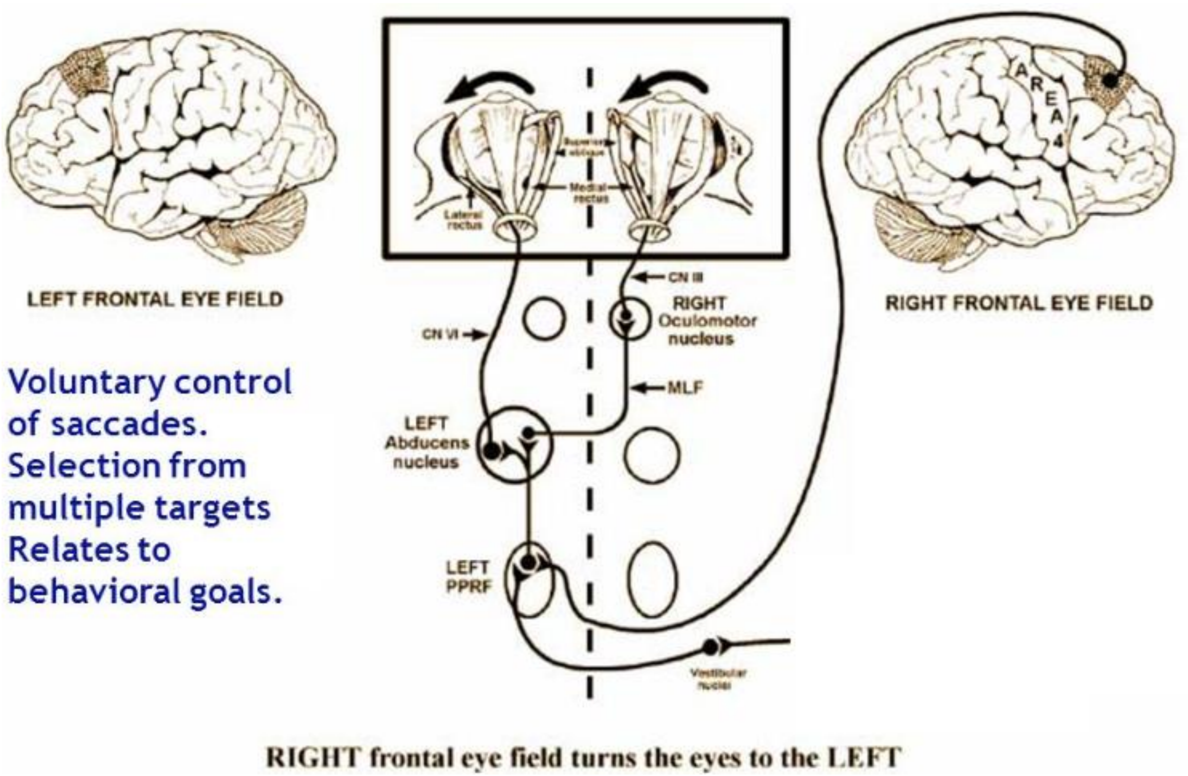
Active Vision

Computer Vision Seminar

Caus Danu

Natural Intelligence



- **Attention** is a **resource**
- Our brain **guides attention** through **eye saccades** (movements)
- We efficiently **sample** the environment and make quick **decisions**

Artificial Intelligence

- Emulate the eye with a **sensor** like: **RGB camera**, **depth** camera, or both (**RGB-D** camera)
- Make our sensor **mobile**
- **Maximize *Information Gain*** (with *minimal effort)*
- **Optimize** sampling **Efficiency**
- **Controll** the point of view of the mobile sensor via a clever **algorithm**

Replace the **eye** with a **camera**

Nonmyopic View Planning

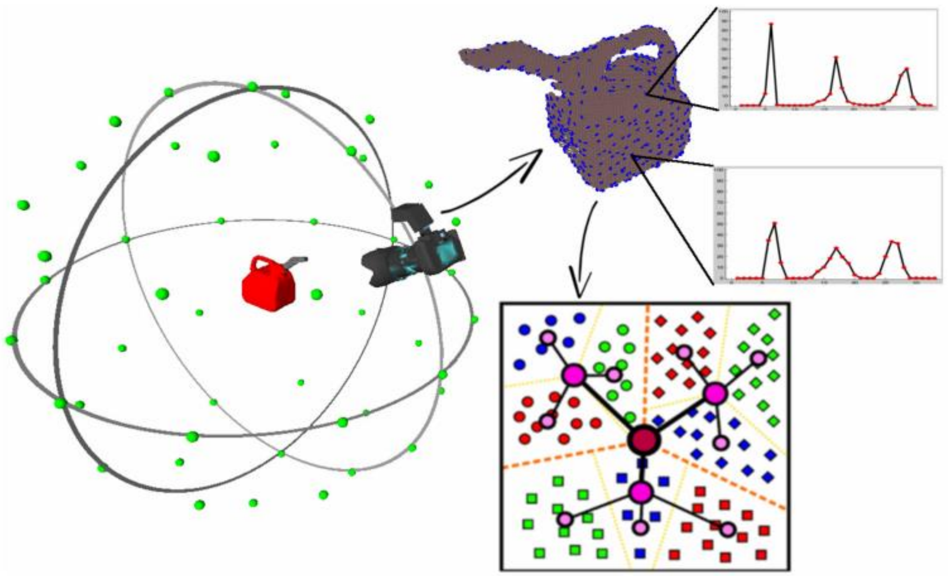
- **Detect and Classify** objects in the scene
- Estimate their **pose**

Bridge the gap

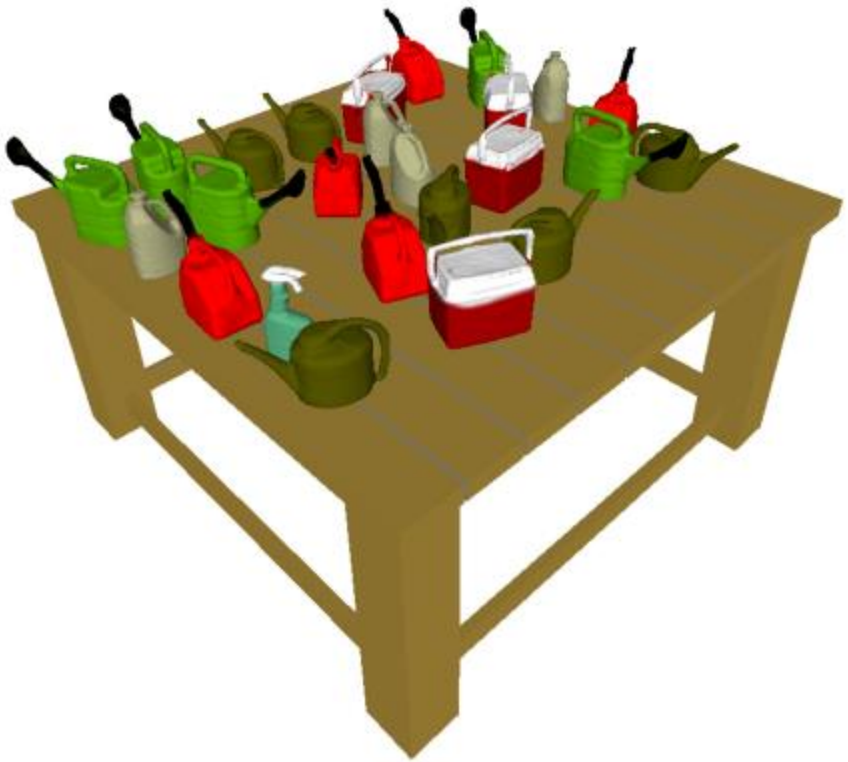
Robot Grasping

- Use a **planner** to estimate a **sequence** of camera poses
- Model using a **POMDP** : **P**artially **O**bservable **M**arkov **D**ecision **P**rocess
- Maximize **long term reward**, i.e. **NOT** Greedy or Short-Sighted/Myopic

Viewpoint-pose Tree



Features used to construct **VP-Tree**



No segmenter component available
(=> Clutter Problem)

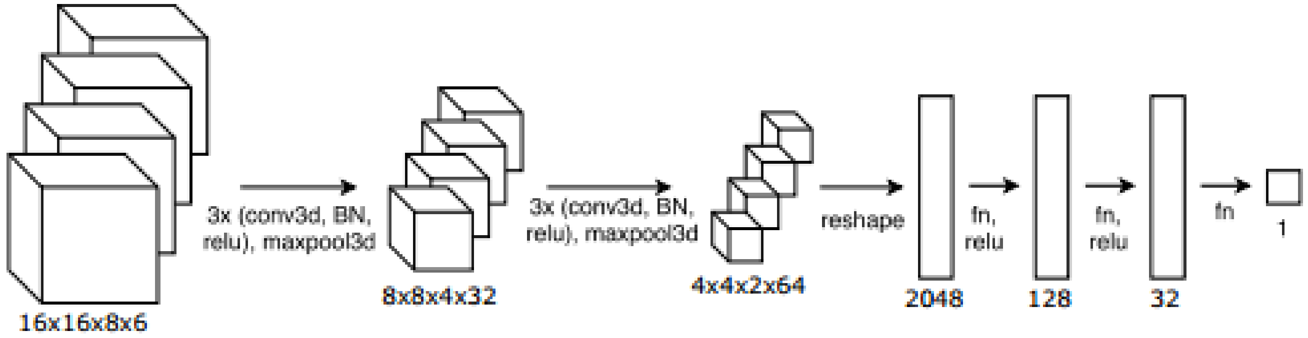
Myopic approach

- **Greedy**/Short-Sighted/Next-Best-View approach
- Care about **short term reward**/one step horizon
- Can be modeled in many ways i.e. MDP, POMDP, ***Neural Nets*** etc.

Especially useful for quadrotor, drone & various flying-robot applications

Oracle Neural Net

- Many Greedy approaches still use a POMDP
- This approach uses a trained **Convolutional Neural Network** that approximates an **Oracle Function**

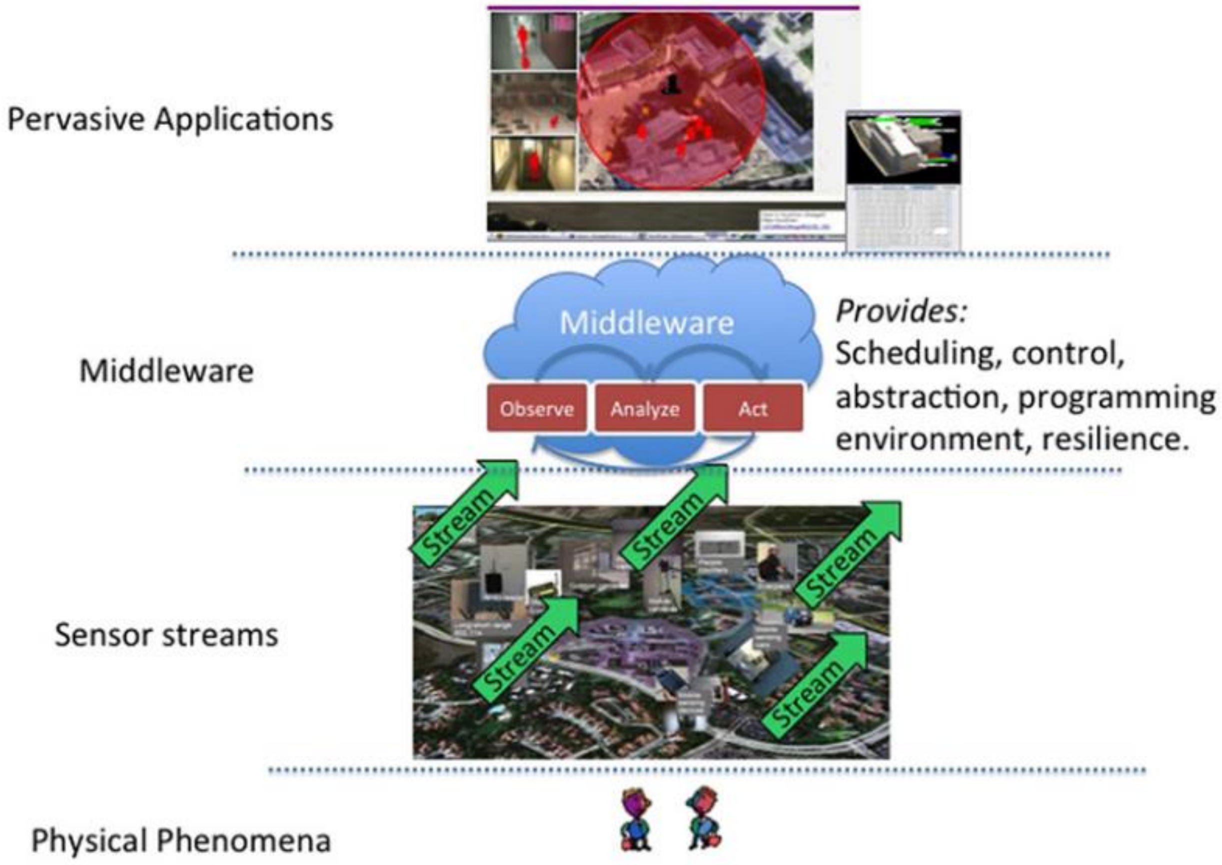


$$\mathcal{L}(X, Y; \theta) = \sum_{i=1}^{N} \|f(X_i) - Y_i\|_2^2 + \lambda \|\theta\|_2^2$$

- Computes **N**ext **B**est **V**iew very fast (inference time of the net is relatively small as compared to a POMDP)
- ***No free lunch !*** Pay with a **long (offline) training time** and lots of **ground truth** data

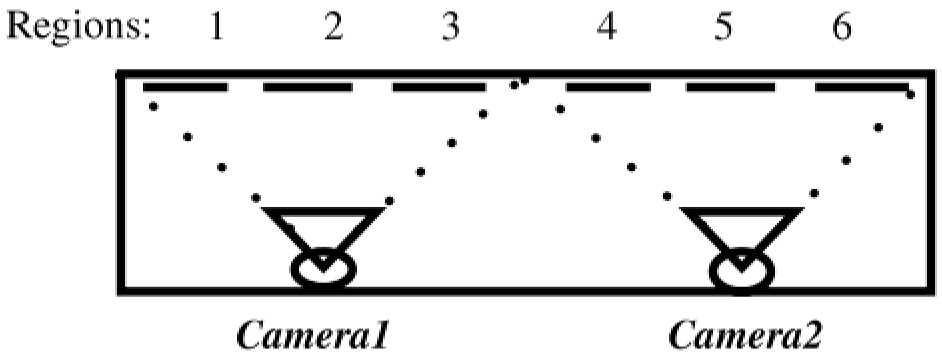
Framework for Monitoring Spaces

- Employ an **approximate POMDP** policy/solution
- Use a **large scale** camera **network** (up to 100 cameras)
- Actuate the pan-**ZOOM**-tilt parameters of cameras

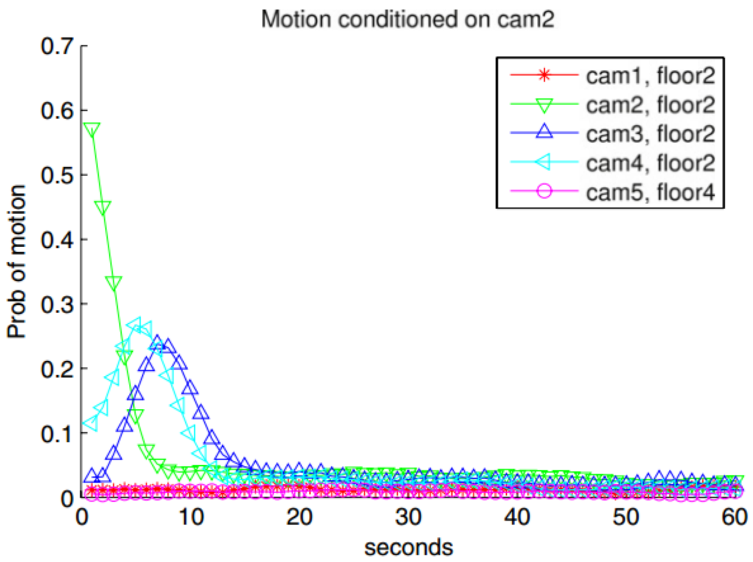


Multiple layers view of **sentient** systems

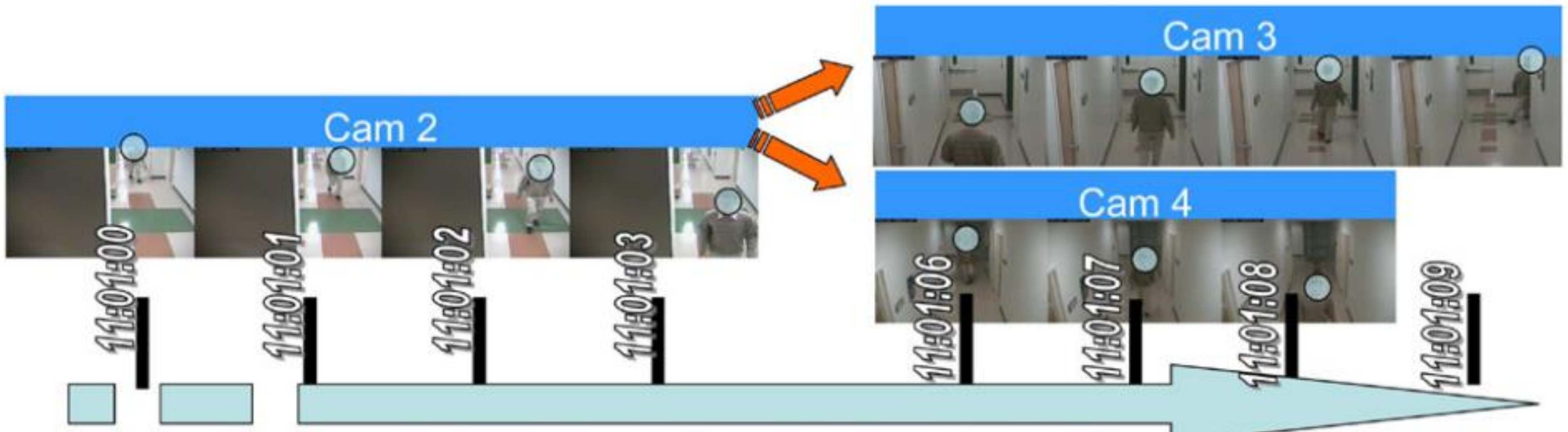
Tips and Tricks



Non-overlapping Fields of View



Extract (correlation-) semantics

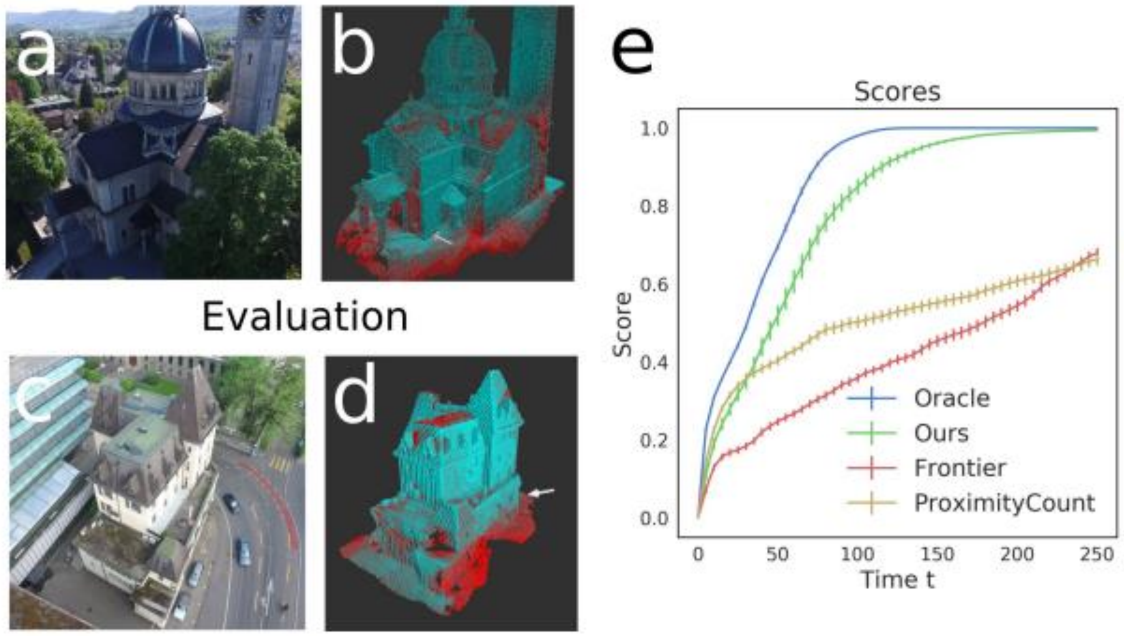


Inject semantics: Look ahead

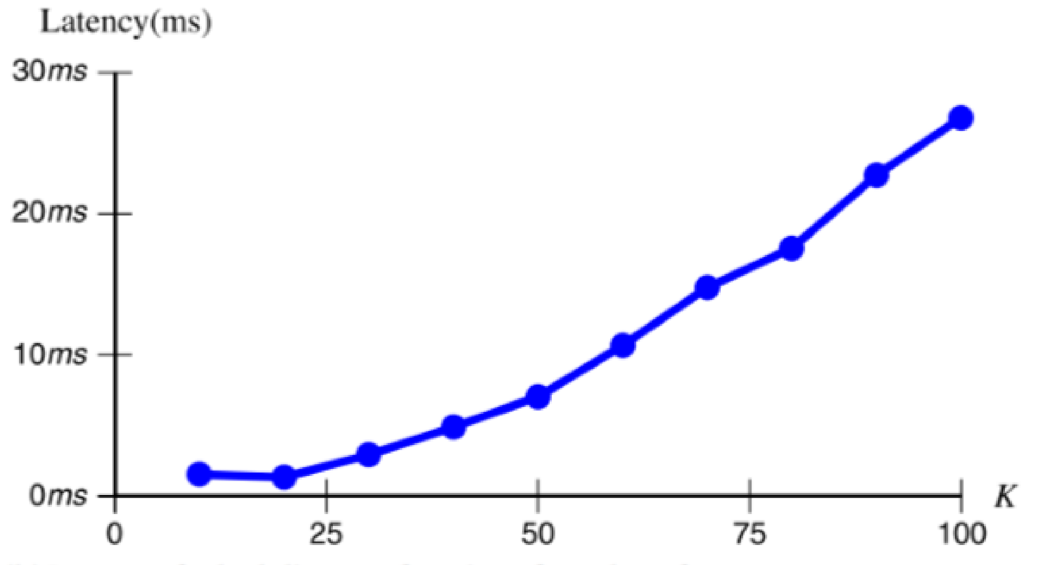
Evaluation and Results

		True Hypothesis							Avg Number of Measurements	Avg Movement Cost	Avg Decision Cost	Avg Total Cost
		H(0°)	H(60°)	H(120°)	H(180°)	H(240°)	H(300°)	H(∅)				
Predicted (%)	H(0°)	87.5	2.5	5.0	0.0	0.0	0.0	5.0	2.53	2.81	9.38	14.72
	H(60°)	2.5	80.0	0.0	0.0	0.0	0.0	17.5	2.66	2.52	15.00	20.18
	H(120°)	7.5	0.0	72.5	0.0	0.0	0.0	20.0	3.16	3.43	20.63	27.22
	H(180°)	0.0	0.0	0.0	70.0	10.0	2.5	17.5	2.20	1.72	22.5	26.42
	H(240°)	0.0	0.0	0.0	2.5	75.0	2.5	20.0	2.39	2.51	18.75	23.65
	H(300°)	0.0	0.0	0.0	0.0	5.0	72.5	22.5	2.57	2.18	20.63	25.38
	H(∅)	0.0	0.0	0.97	0.0	0.0	0.97	98.05	2.17	1.93	1.46	5.56
											Overall Average Total Cost:	**20.45**

NVP (Nonmyopic View Planner) Approach (on PR2 robot)



Myopic CNN Approach



Latency of scheduling as a function of number of cameras

Approximate POMDP Approach

Concluding Remarks

1) **NVP** approach works well in simulation and we can **transfer the encoded knowledge** to a real robot

   - It is not much better quantitively than a state of the art **Greedy Approach**, but the main advantage is the **adaptive stopping criterion**

   - Does not scale well to **large landscapes,** since solving POMDPs is **exponentially complex**

   - Would benefit from a **custom segmenter** to avoid unnecessary camera movements

2) **CNN** approach works much better for **large scenes**

3) We can still leverage POMDPs for large scale systems by creatively **engineering clever setups**, employing **simplifying assumptions** and reasonable **approximations**

“Always ask questions for an Active Learning“

Active

LEARNING