**INTRODUCTION**

- Autonomous MAVs require accurate perception to successfully navigate, avoid collisions, and make decisions [1] in real-time.
- RGB cameras provide high-resolution data at fast rates in a cheap and lightweight sensor package.
- Passive vision-based 3D reconstruction algorithms typically utilize either Structure from Motion [2] or Simultaneous Localization and Mapping [3].
- Active methods that control the sensor to minimize the reconstruction error, e.g., Next Best View or active target tacking, often operate in discretized spaces with specific system models.
- Our control algorithm actively seeks the most valuable camera motions for estimating the 3D positions of image features in the local robot frame [4].
- We derive explicit expressions for the gradient of the error covariance for both the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) at the next step for a general nonlinear system.
- We further propose a simple heuristic to drive a camera in parameterized circles to mimic the gradient-based methods.
- Active sensing methods are demonstrated on a quadrotor with a downward facing camera in both simulations and hardware experiments.

**ACTIVE VISION SOLUTION**

- General nonlinear dynamics and measurement model:
  \[
  x_{k+1} = f(x_k, u_k) + v_k \quad \text{Measurements: } y_k = h(x_k) + w_k \quad \text{Control: } u_k \in \mathbb{R}^n
  \]
- Definitions:
  - \( p_k := p(x_k | y_{1:k}) \)
  - \( \Pi_k := E(x_{k+1} | y_{1:k}) = E(x_k | y_{1:k}) + \mathcal{N}(0, P_{k+1}) \)
- Minimizing expected value of a function of the probability at some future time, i.e., \( J(P_{k+1}) \), leads to an intractable POMDP.
- We instead minimize the cost of the one-step future estimation error covariance:
  \[
  J(P_{k+1}) = 
  \begin{aligned}
  &\text{cost of one-step future estimation error covariance: } J(P_{k+1}) = \\
  &\text{Control found by solving for } u_k \text{ at each time-step from: } \\
  &\text{Final desired control: } u_k^* = \Gamma \gamma_k
  \end{aligned}
  \]
- Closed-form EKF gradient:
  \[
  \begin{aligned}
  \Pi_k &:= \mathcal{F}(x_k, u_k) + Q \\
  \Sigma_k &:= \mathcal{F}(x_k, u_k) + G_k \Sigma_{k+1} G_k^T + R \\
  x_{k+1} &:= x_{k+1} + G_k \Sigma_{k+1} \Pi_k \\
  y_{k+1} &:= y_{k+1} \quad \text{Error covariance:
  for one image feature: } \\
  y_k = \frac{1}{2} K x_k + w
  \]
- Mobile pinhole camera system
  \[
  x_{k+1} = \begin{bmatrix}
  x_k \\
  y_k \\
  z_k
  \end{bmatrix}
  \quad \text{for one image feature: } \\
  y_k = \frac{1}{2} K x_k + w
  \]
- Multiple feature version uses solution from over-determined system

**SIMULATION RESULTS**

- Simulation environment in MATLAB models virtual pinhole camera onboard a quadrotor with full nonlinear dynamics.
- Quadrotor control (200Hz): 1) Velocity PID control with desired velocity and 2) Attitude PID control [5].
- Performed 100 Monte Carlo simulations with random initial conditions to determine active trajectories and propose active estimation proxy (Fig. 1).

**EXPERIMENTS**

- **Quadrotor:** Komet K500 (Fig. 4).
- **Computation:** Odroid XU-4 micro-computer (Ubuntu 14.04, ROS, OpenCV 2.4.3, 2Gb ARM processor, 2GB RAM, 64GB storage).
- **Communication:** ROS Indy handles image acquisition, feature tracking, filtering, and active control computation.
- Actively estimated 10 image features that represent the 3D object at 15Hz.
- Active EKF (Fig. 5) outperforms random walk and greedy selection.

**CONCLUSION**

- Proposed closed form, computationally efficient active perception control for general nonlinear systems for EKF and UKF.
- Applied method to specific 3D reconstruction with monocular camera on quadrotor in simulations and experiments.
- Active estimation methods are closely approximated by simple active circular controller for the quadrotor scenario.
- Future work includes multi robot extension and more complex applications.

**ACKNOWLEDGEMENTS**


**REFERENCES**


---

*Active Vision-based Perception for Fast 3D Reconstruction with an Aerial Robot*

*Active Vision-based Perception for Fast 3D Reconstruction with an Aerial Robot*