coggle
made for free at coggle.it

Drone Racing

- **Human Pilot**
  - Competitive control skills require years of training.
    - Valuable for autonomous systems which must fly safely and quickly through complex environments, i.e. disaster response, aerial delivery, inspection of complex structures).
  - Radio Transmitter
- Drone Racing Sports
  Video
  - Radio Transmitter
  - Forward-Facing Onboard Camera
- **Drone**
  - Forward-Facing Onboard Camera
- **Race Track**
  - Static
    - Gates/Obstacles
  - Dynamic
    - Tight Coupling of Perception and Control
      1~[7,28]
      - Maintain Visual Contact to Gate
      - Look for Gate
      - Localize Relative to Gate
    - Designed for predominantly static worlds
    - Static Waypoints and Trajectories
- Autonomous Systems
  - Challenges
    - Optimal Control
    - Localization and Mapping
    - Dynamics Modelling
    - Onboard Perception
    - Trajectory Generation
  - **Visual Servoing**
    Traditional Solution
    - Set of target locations in the form of reference images 1~[30]
    - Problems
      - Difference between current and reference image must be small
        - Motion Blur
        - High Speed
        - Occlusions
  - **Gate Detectors**
    Traditional hand crafted
    - Problems
      - Occlusions (=Partial Visibility)
      - Motion Blur
  - **Global Trajectory and State Estimation**
    State estimation must be accurate.
    - Challenges
      - Designed for predominantly static worlds
      - Severe Drifts at High Speed
        - Perceptual Aliasing
        - Motion Blur
        - Lighting Conditions
      - Expensive Sensors 1~[2]
      - High Computational Costs 1~[22]
        - Compressed maps lead to drift 1~[18]
        - Real-time performance onboard resource constrained platforms, i.e. small quadrotors
          - Difficulties in exploiting popular minimum-snap or -jerk trajectories 1~[19,20]
    - **SLAM**
      Simultaneous **L**ocalization **A**nd **M**apping
      1~[3]
      - Accurate pose estimates against a globally consistent map which was previously-generated.
      - **Challenges**
        - High Computational Demands
          - Real-time performance onboard resource constrained platforms, i.e. small quadrotors
        - High Accelerations and Speeds
          - Motion Blur
          - Loss of Feature Tracking
        - Map created under different conditions
      - **Limits**
        - Predominantly static environments
          - Static Waypoints and Trajectories
  - **Perceptive Systems**
    - **Vision Based Systems**
      - **Hybrid ML System**
        perceptual awareness of a (CNN) with speed and accuracy of a state-of-the-art trajectory generation and tracking pipeline
        - **Experiments**
          - **Simulation**
            - Output
              - Higher Robustness and Precision than state-of-the-art, highly engineered systems
              - Compactness for Onboard Execution
              - Highly Dynamic Environments
              - Severe Occlusions (=Partial Visibility)
              - Complex Navigation Tasks (seeking moving gates, racing though tracks)
          - **Real World**
            - Output
        - **Subsystems**
          - **Perception with CNN**
            - Able to reason on high-dimensional raw sensory input
            - Design
              - Loss Function - Weighted MSE loss on point and velocity predictions $L = ||\vec{x} - \vec{x}_g||^2 + \gamma(v - v_g)^2$
                - Shown with Cross Validation
                  Optimal weight: $\gamma = 0.1$
                  but performance mostly insensitive to weight
              - ResNet-based architecture of Loquercio et al. [17]
                ~ efficient for onboard
              - Performance
                - 10 Frames per Second
            - Data Collection
              - Variety of race track layouts
                - Jointly Training
              - Simulation
                - Variant of DAgger 1~[26]
                  - Recovering when learned policy deviates far from the reference trajectory
              - Dataset
                - State Estimates + Corresponding Camera Images
                  - Ground Truth with Expert Policy
              - Real World
                - Manually Carrying the quadrotor through the track
                  !Sufficient Coverage of Off-Trajectory Positions
            - Training
              Imitation Learning
              - Network is not directly imitating the demonstrated behavior. Thus, performance/abilities of learned policy are not upper-bounded by performance/abilities of provided demonstrations
              - Jointly Training
                - *Generalization to Dynamic Environments*
                  Increased Robustness
              - Supervision
                - Pre-computed Global Trajectory t_g
                  Minimum Snap 1~[19]
                  - Requirements at Training Time
                    - Location of Gates in Reference Frame
                    - Accurate State Estimates of Quadrocopter in Reference Frame
              - Expert Policy
                - Quadrotor Position $\vec{p}_c \in R^3$
                  - Closest Point on t_g: $\vec{p}_c'$
                    - distance{ $\vec{p}_c, \vec{p}_g$ } = Prediction Horizon $d \in R$
                      → Desired Position $\vec{p}_g$
                      - Project $\vec{p}_g$ onto image plane
                        → Ground Truth Normalized Image Coordinates $\vec{x}_g$
                        - Normalize Speed at $\vec{p}_c'$ with maximum speed along t_g
                          → Desired Speed v_g
            - Learned Policy
              - Input
                - **ONLY** Raw Image from facing-forward onboard camera
                  in the paper: 300x200 RGB image
              - Output
                - Waypoint in Local Image Coordinates $\vec{x} \in [-1, 1]^2$
                  Normalized Desired Speed $v \in [0, 1]$
          - **Path-Planning**
            - Input
              - Waypoint in Local Image Coordinates $\vec{x} \in [-1, 1]^2$
                Normalized Desired Speed $v \in [0, 1]$
              - Transfer Procedure
                - **Conversion of $\vec{x}$ to 3D local frame coordinates**
                  $\vec{p}_g = [p_{g1}\ p_{g2}\ p_{g3}]$
                  - *Set prediction horizon* d *proportional to* v
                    Set depth $p_{g1} = d$
                  - *Back-project* $\vec{x}$ *along camera projection ray*
                    $\rightarrow p_{g2},\ p_{g3}$
                - Conversion of v to desired quadrotor speed v_{des}
                  $v_{\text{des}} = v_{\max} \cdot v$
                  $v_{\max}$ *user-defined, controls aggressiveness of flight*
                  - Minimum-Jerk Trajectory 1~[20]
                    computationally efficient
            - Output
              - Minimum-Jerk Trajectory Segment t_s
                1~[20]
          - **Low-Level Control**
            - Input
              - Minimum-Jerk Trajectory Segment t_s
            - Track t_s with control scheme by Faessler et. al. 1~[15]
            - Output
              - Motor Commands
                Low-Level Control Commands
        - **Abilities**
          - Speed and Accuracy of trajectory generating and tracking pipeline
          - Simplicity of CNN
          - Test: Precision and Robustness
          - Real Time Application
          - Autonomous Flight through race track with moving gates with high agility
          - Onboard Computation
          - Agile Flight Trajectories
          - Dynamic Environments
            - Moving Gates
          - No explicit map of environment required
          - Evaluation against:
            State-of-the-Art Navigation Methods
            Professional Human Drone Pilots
          - Simulation/Real World
      - **End-to-end ML Systems**
        Learning-based algorithms that couple perception and control
        - Motor commands directly derived from images. 1~[25,27,14,4,24,13,11]
        - Abilities
          - Independent of Global Maps
          - Independent of Position Estimates
        - Shortcomings
          - Inherent difficulties of generalizing to 3D motions 1~[25,4,24]
          - High computational complexity 1~[27,11]
          - Low maximum speed 1~[13]
        - Open Questions
          - Optimal Output Representation
            - Discrete Navigation Commands 1~[15,17]
              - + High Robustness
              - - Agility
            - Direct Control [14]
              - + Agile control
              - - High Sample Complexity