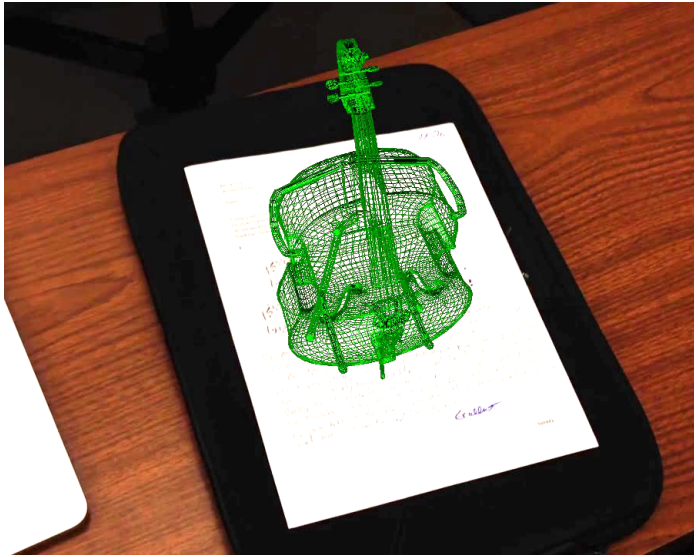


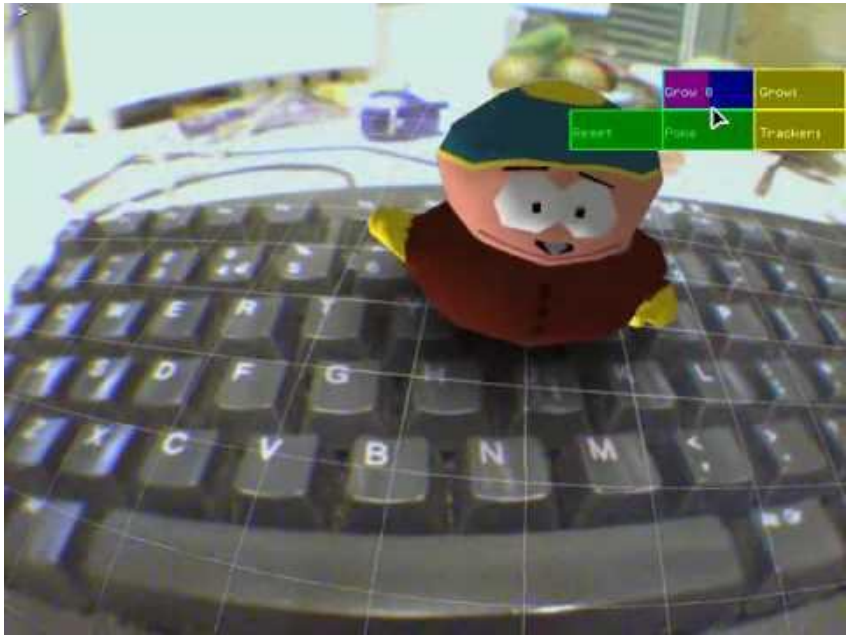
Motion Tweening

Kevin Yeh, Matt Broussard, Kaelin Hooper, and Conner Collins

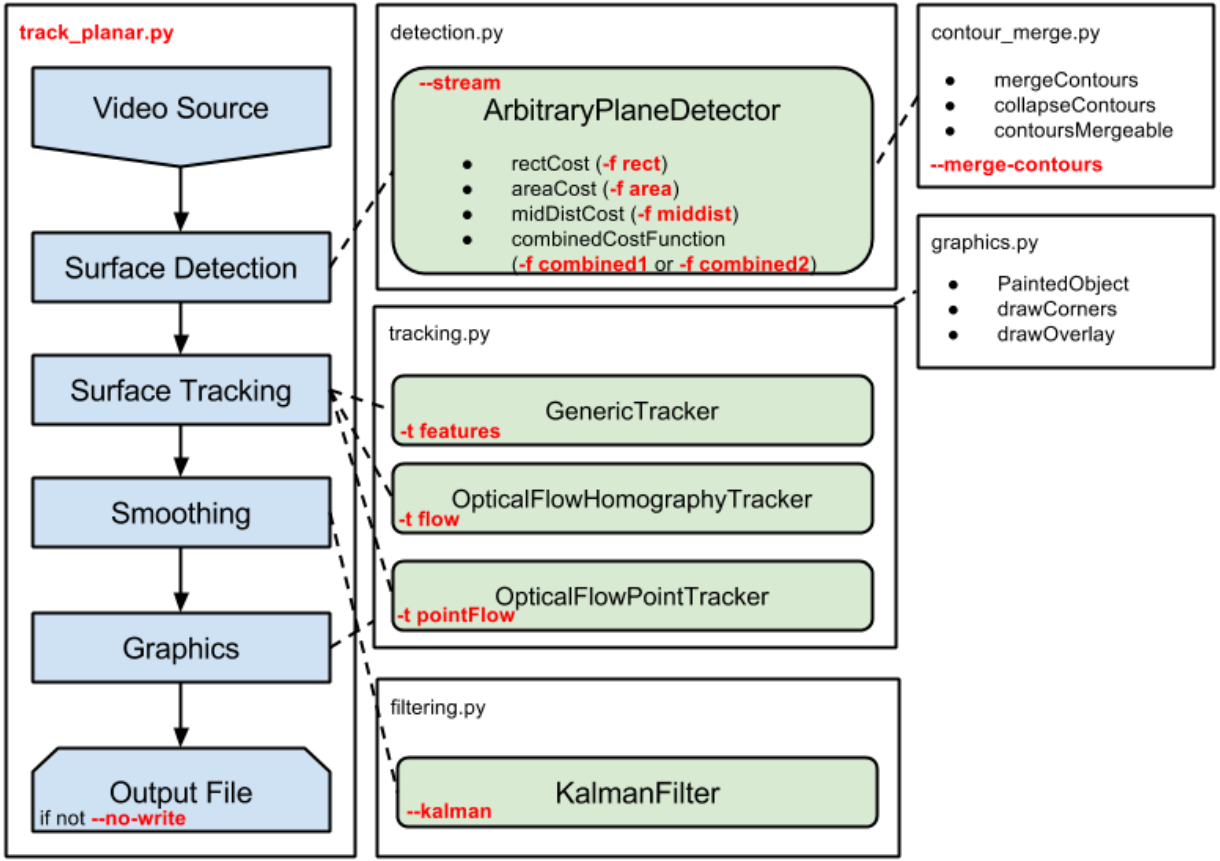
Overview

- Goals:
  - Track a planar surface such that the perspective of a 2D or 3D object changes with the perspective of the surface.
- Inspiration:
  - Klein-Murray PTAM Paper
  - Disney’s Paperman Short

Inspiration



Methods

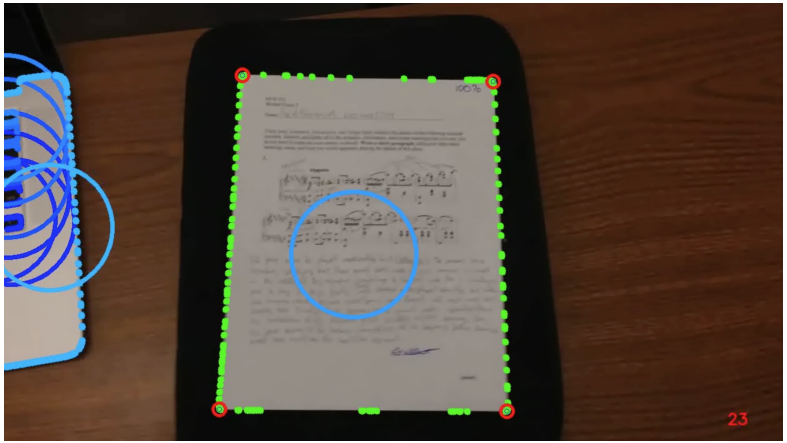

track_planar.py
Video Source
Surface Detection
Surface Tracking
Smoothing
Graphics
Output File
if not --no-write
detection.py
--stream
ArbitraryPlaneDetector
rectCost (-f rect)
areaCost (-f area)
midDistCost (-f middist)
combinedCostFunction
(-f combined1 or -f combined2)
tracking.py
GenericTracker
-t features
OpticalFlowHomographyTracker
-t flow
OpticalFlowPointTracker
-t pointFlow
filtering.py
KalmanFilter
--kalman
contour_merge.py
mergeContours
collapseContours
contoursMergeable
--merge-contours
graphics.py
PaintedObject
drawCorners
drawOverlay

Planar Surface Detection

- Originally, required training image.
- New Process:
  - Detect contours with Canny edge detection
  - Merge contours that appear to compose rectangles
  - Choose one contour using cost function minimization
  - Quadrilateral Corner Estimation

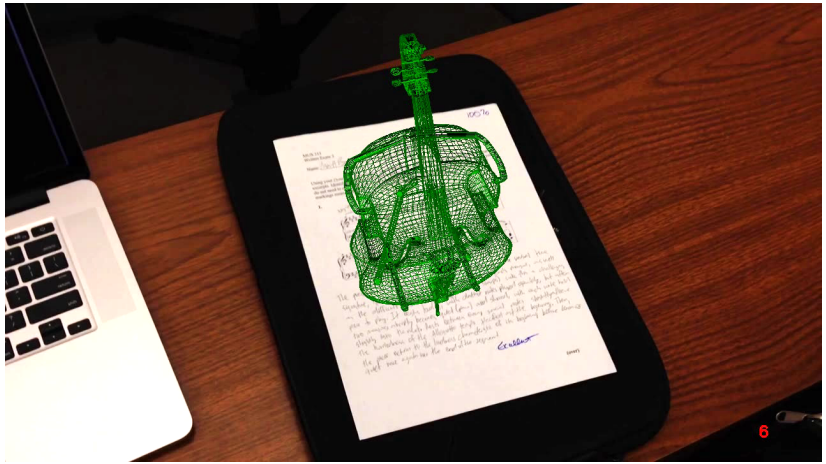
Surface Tracking

- Overview:
  - SIFT Feature Matching
  - Sparse Optical Flow
  - Dense Optical Flow
  - Naive (Continuous Detection)



Model Mapping

- Model Normalization
- Mapping



- Also allow user paint interface

Results & Demonstration

- Videos
  - Plain White Page
  - Textured White Page
  - User Paint on Star Wars Robot
  - User Paint on Animation
  - 3D model on Longhorn T-Shirt

Challenges

- Corner intuition from colinearity
- Ordering of contours list
- Difficult to calibrate weights for cost function
- Kalman filter ineffective
- Poor OpenCV python documentation :’(

Thank You

Questions?