**Motivation**

- Explore the technology behind adding content to a live video in real-time
- Detect and track “fiducial” markers in video stream, estimate homography and replace with artificial imagery

**Marker Detection**

- Simple SIFT/SURF feature matching between marker image and video
- Pre-compute marker keypoints/descriptors
- Match with descriptors on video frames
- Filter matches with distance ratio test and RANSAC
- Compute homography between marker image and video frame (could estimate 3D with camera extrinsics)

**KL Tracking**

- **KL Tracking** of keypoints between frames:
- Estimates displacement (ignore affine transformation) of small windows around keypoints by minimizing the error between two consecutive frames
- Stop tracking keypoints when minimized error is too high

\[ \varepsilon = \int_{w} [J(Ax - d) - I(x)]^2 w \, dx \]

- Let A=I, approximate I(x) with truncated Taylor series
  \[ I(x - d) = I(x) - g \cdot d \]

- **Affine Consistency Check**: Estimate displacement AND affine transform between first and current frame
  - Stop tracking keypoints when minimized error is too high
  - Can be computed iteratively by solving system of equations

**Results/Comments**

- SIFT is slower than 30fps
- KL Tracking operates well on textured windows and corners
- SIFT keypoints (blobs) not the best for tracking, OpenCV’s goodFeaturesToTrack() is better BUT they are harder to match
- Tracking is imperfect, even with OpenCV’s implementation of KL tracking. Maintaining correspondence for homography is hard
- Other tracking methods, such as per-frame contour-based recognition might be better suited for binary fiducial markers

**References**