Real-Time Detection and Tracking for Augmented Reality on Mobile Phones

D. Wagner, A. Mulloni and D. Schmalstieg

Presented by
Yuansi Chen and Lingqi Yan
Outline

1. Motivation and Related Work
2. Modified Features Detectors
3. Performance and Analysis
Motivation
Motivation

• Limited computational resources (speed and memory) on Mobile devices
• Natural feature tracking infeasible: SIFT and Ferns
Goal

• Enough speed improvement for real-time AR processing
• with Limited memory
• without losing too much quality
• on real phones (<33ms/frame)
Related Works

• General Feature Detectors for PCs (slow)
• Outsource the tracking task to PCs via wifi. (AR-PDA project: 10s per frame is still slow)
• Marker tracking: restricted applications
Detection and Tracking Routine

- Keypoint Detection
- Feature Matching
- Outlier Removal
- Pose Estimation and Refinement

Activate when lost

Activate if target is found and sure

Local-Search-Based Tracker
Scale Invariant Feature Transform (SIFT)
Ferns

- Feature detection as classification
- Binary Feature $F(p)$
- $C = \text{argmax} \ P(C_i|F)$
- Instead of storing full joint distribution, add independence:
  \[
P(F|C) = \prod P(F_S|C)
\]
FAST Corner Detector

Ref from:
<Machine learning for high-speed corner detection>
By Edward Rosten and Tom Drummond, University of Cambridge

• **Features from accelerated segment test (FAST)**
  
• A corner detector many times faster than DoG but not very robust to the presence of noise
• Based on intensity level tests
FAST Corner Detector
SIFT to PhonySIFT

Main Modifications:
• Uses FAST corner detector to all scaled images to detect feature points instead of scale-crossing DoG
• Only 3x3 subregions, 4 bins each, creates 36-d vector
• Using a Spill tree
Ferns to PhonyFerns

Main Modifications

• Uses FAST detector to increase detection speed
• Reduces each ferns size
• Uses 8-bit size to store probability instead of using 4 bytes float point value
• modifying the training scheme to use all FAST responses within the 8-neighborhood
Outliner Removal

• Orientation Estimation

• Homography verification based on RANSAC/PROSAC
PatchTracker

Ideas:
1. Both the scene and the camera pose change only slightly between two successive frames
2. New feature positions can be successfully predicted by old one with defined range search
Combined Tracking

- Recognition
  - Keypoint detection

- FERNS
  - Image blur
  - Classification
  - Outlier removal

- SIFT
  - Descriptor creation and matching
  - Outlier Removal

- Pose estimation and refinement

- Tracking
  - PatchTracker

Activation:
- Activate PatchTracker if target was found
- Activate PhonySIFT/PhonyFerns if target was lost
Performance & Analysis

• Platform: Asus P552W (Cellphone)
  – 624Mhz CPU
  – 240x320 screen resolution
  – No float point unit
  – No 3D acceleration

• Platform: Dell Notebook (PC)
  – 2.5Ghz , limited to use single core
  – With float point support
Speed

Performance on the phone:
- PhonySIFT: 38.3 ms per frame
- PhonyFerns: 41.3 ms per frame
- PhonySIFT with PatchTracker: 8.4 ms per frame
- PhonyFerns with PatchTracker: 8.3 ms per frame

Performance on the PC:
- PhonySIFT: 3.8 ms per frame
- PhonyFerns: 3.2 ms per frame
- PhonySIFT with PatchTracker: 1.0 ms per frame
- PhonyFerns with PatchTracker: 1.0 ms per frame
Robustness over different objects

Fig. 5. The seven test sets (a)-(g): book cover, advertisement, cars movie poster, printed map, panorama picture, photo, and Vienna satellite image.
Robustness over different objects
Typical Situations of Switch
Typical Situations of Switch

• (Show paper Figure 7)
Ferns vs SIFT vs PatchTracker

- Simple - Pixel errors
- Simple - Inlier count
- Tilt - Pixel errors
- Tilt - Inlier count
- Occlude - Pixel errors
- Occlusion - Inlier count
- Fast Movement - Pixel errors
- Fast Movement - Inlier count
- Lose Target - Pixel errors
- Lose Target - Inlier count
Detailed Speed Analysis

• **PhonySIFT:**
  - Corner Detection (FAST) : ~14%
  - Feature descriptor and Matching : ~74%
  - Outlier Removal : ~ 9%
  - Pose Refinement : ~ 3 %

• **PhonyFerns:**
  - Corner detection (FAST) : ~ 22%
  - Second Octave and Blurring : ~ 17%
  - Classification : ~ 59%
  - Outlier Removal : ~ 2%
Conclusion

- Successfully worked with tracking system on phones
- Better CPU would come out in the future. The choice of the next generation feature is unknown