TRICTRAC: Target detection and tracking using feature points.

T. Mathes

Université de Liège

Multitel Workshop 08/06/2006
Outline

1. Introduction
2. Detection
3. Tracking
Outline

1 Introduction
2 Detection
3 Tracking
Challenges and problems

Challenges:

- **Occlusions**: targets appear, disappear
  → Robust tracking

- **Proximity of similar targets**
  → Use interest points and local descriptors

Problems:

- **Fast motion, blur**: local features will tend to disappear
  → use multi-scale feature detection
Challenges and problems

Challenges:
- **Occlusions**: targets appear, disappear
  - Robust tracking
- **Proximity of similar targets**
  - Use interest points and local descriptors

Problems:
- **Fast motion, blur**: local features will tend to disappear
  - Use multi-scale feature detection
Challenges and problems

Challenges:
- **Occlusions**: targets appear, disappear
  → **Robust tracking**
- **Proximity of similar targets**
  → Use **interest points** and local descriptors

Problems:
- **Fast motion, blur**: local features will tend to disappear
  → use **multi-scale feature detection**
Challenges and problems

Challenges:

- **Occlusions**: targets appear, disappear
  → Robust tracking

- **Proximity of similar targets**
  → Use interest points and local descriptors

Problems:

- **Fast motion, blur**: local features will tend to disappear
  → use multi-scale feature detection
Outline

1. Introduction
2. Detection
3. Tracking
The challenge

- Traditional background subtraction (e.g. mixture of gaussians) is inappropriate here.
- How to handle a moving camera?
- How to cope with fast and unexpected illumination changes?
Our approach

- Detect non-static points
Our approach

- Group them by using connected components.
Results

- Static camera: Detection, Detection (Map)
- Static camera: Detection.
- Moving camera: Detection, Detection (Map)
Effect of illumination changes

TRICTRAC: Target detection and tracking using feature points.
Combining Object Detection with Tracking

- Mask tracked objects to avoid re detections.
- Tracking, Tracking (Map)
Problems

- The method is very ad-hoc. Can we do better?
  → Use a Kalman filter for each feature point.

- Sometimes the feature points are too sparse to have good detections.
  → Use a better feature point detector (features and blobs); use graph morphology to fill in the holes.
Problems

- The method is very ad-hoc. Can we do better?
  → Use a Kalman filter for each feature point.

- Sometimes the feature points are too sparse to have good detections.
  → Use a better feature point detector (features and blobs); use graph morphology to fill in the holes.
Problems

- The method is very ad-hoc. Can we do better?
  → Use a Kalman filter for each feature point.
- Sometimes the feature points are too sparse to have good detections.
  → Use a better feature point detector (features and blobs); use graph morphology to fill in the holes.
Kalman filter on each feature point

- Each new point generates a Kalman filter.
- Points are matched to nearby filters from frame to frame.
- If no measurement (point dissappears), the uncertainty increases.
- If uncertainty gets very large (no corresponding point for a long time), filter is removed.
- To handle static and non-static points, we can switch between zeroth and first order Kalman filter.

Detection, Detection (Filters)
I’m currently trying to integrate this into my detection and tracking framework.

- Compute the Hessian:

\[
H(x, \sigma) = \begin{pmatrix}
L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\
L_{xy}(x, \sigma) & L_{yy}(x, \sigma)
\end{pmatrix}
\]  \hspace{1cm} (1)

- Detect local maxima of \(L_{xx}L_{yy} - L_{xy}^2\) \(\rightarrow\) features
- Detect local maxima of \(L_{xx} + L_{yy}\) \(\rightarrow\) blobs
- Features are more dense and more stable, and are more robust to blur and fast motion.
Filtering

SURF results

SURF results
Outline

1. Introduction
2. Detection
3. Tracking
Different Tracking Approaches

- Background subtraction
- Contour-based approaches
- Feature-based approaches (TRICTRAC)
- ...
Static Model vs. Adaptive Model

- Static model will loose object if object appearance changes (due to illumination, scale changes, i.e. anything not learnt by the model)
- Adaptive model is subject to drift (can start tracking something else...)
- A mix of a priori information and online updating could help...
• Extract sparse sets of salient image points using Harris corner detector.

• Local appearance:

\[ \mathbf{v} = (x\ y\ r\ g\ b\ \mathbf{r}_x\ \mathbf{r}_y\ \mathbf{g}_x\ \mathbf{g}_y\ \mathbf{b}_x\ \mathbf{b}_y)^T \]

where \( \mathbf{v} \in \mathbf{V} \) with \( \mathbf{V} \subset \mathbb{R}^{11} \).

• Advantages of interest points:
  save computation time and improve robustness.
Constructing the Model

- **Shape definition:**

  \[ X = \begin{pmatrix} v_1^T & v_2^T & \ldots & v_N^T \end{pmatrix}^T \]

  where \( X \in M \) with \( M \subset \mathbb{R}^{11N} \).

- In general, \( M \) is a non-linear manifold.

- Collect a set of shapes.
Constructing the Model

- Assumption 1: Small deformations, so $\mathbf{M}$ is linear.
- Assumption 2: Distribution of shape vectors Gaussian.
- Due to correlations between interest points, $\mathbf{M}$ is low-dimensional.
Constructing the Model

- This gives a model with a low number of deformation parameters.
- Using the model, similar shapes can be generated.
Examples: Training, Tracking 1, Tracking 2, Tracking 3.

- Cannot learn large deformations. (linearity of manifold only if small deformations)
- Cannot handle appearance changes. (like out-of-plane rotations)
- The whole trick is to find out how to do this...
Model can incorporate *new points* and remove *inactive points*.

\[ n \quad n + 1 \quad n + 2 \quad n + 3 \quad n + 4 \]

**Observation**

**Model**
Tracker Properties

- The current tracker can cope with non-rigid objects, out-of-plane rotations, severe partial occlusions, scale changes and illumination changes.
- No background subtraction is required.
- The tracker can start tracking in the first frame; no initial training sequence required.
- Automatic track termination if target lost.
Latest Tracking Results

- Example 1 (Soccer)
- Example 2 (Soccer)
- Example 3 (Video surveillance)
Next steps

- Integrate SURF into detection and tracking.
- Combine tracker with particle filter.
- Make tracker more probabilistic (each point of the model should have a confidence score) → this should increase robustness and ease integration of new and elimination of bad points.