Computer Vision for HCI

Covariance Tracking

1

Covariance Tracking

Based on: “**Covariance tracking using model update based means on Riemannian manifolds**”,
Fatih Porikli, Oncel Tuzel, Peter Meer , CVPR 2006

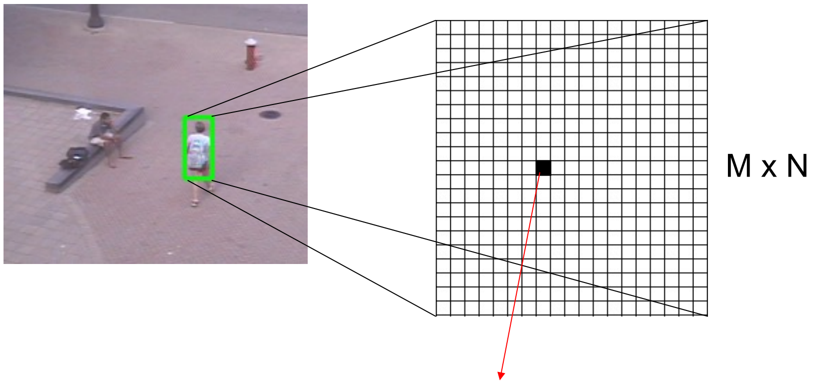
2

Motivation

- Capture spatial and statistical properties, **and their correlation**
- Can fuse different types of features
  - Location
  - Color
  - Edges
  - Motion
- Low dimensional representation (fast)
- Scale invariance

3

Covariance Descriptor

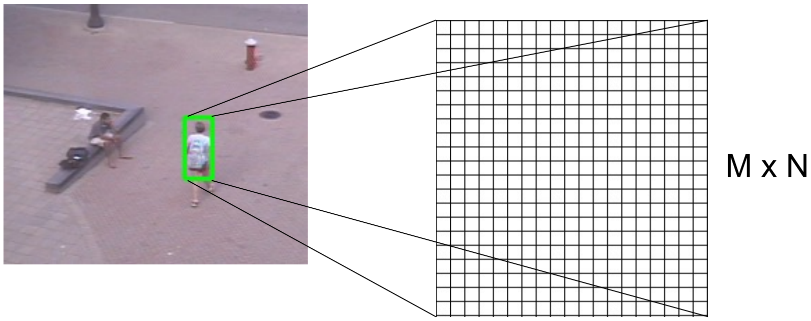


Feature vector (for each pixel) $f_k = [\ x\ y\ R\ G\ B]^T$

d x 1

4

Covariance Matrix



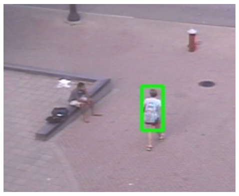
Person Model

$$C^{Model} = \frac{1}{MN}\sum_{k=1}^{MN}(f_k-\mu)(f_k-\mu)^T$$

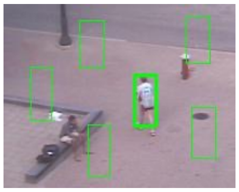
d x d

Mean of all f_k

Finding the best match



C^{Model}



Candidate matches

- For each possible patch region in the <u>next</u> image, find its covariance matrix
- Compare C^{Model} to the covariance matrices of all possible patch regions in next image
- Find the patch in next image whose distance from C^{Model} is <u>minimum</u>

Comparing Covariance Matrices

- Find the distance between 2 covariance matrices
- Space of covariance matrices is <u>not</u> vector!
  - Simple arithmetic matrix subtraction would <u>not work</u>
- The space of covariance matrices is a **Riemannian Manifold**
- Distance metric:

$$\rho(C^{Model}, C^{Candidate}) = \sqrt{\sum_{i=1}^{r} [\ln(\lambda_i(C^{Model}, C^{Candidate}))]^2}$$

λ_i - *Generalized* Eigenvalues [In Matlab: *eig(C$_1$,C$_2$)*]

$$\sqrt{\sum_{i=1}^{r} [\ln(\lambda_i(C^{Model}, C^{Candidate}))]^2}$$

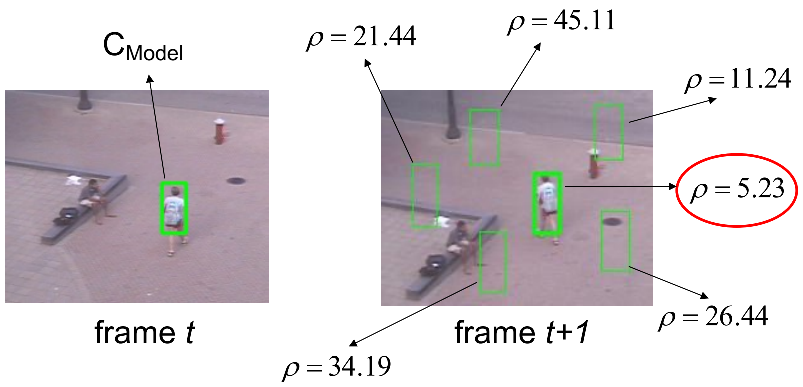
Algorithm

- Compute C^{Model} for known target in <u>current</u> image
- Scan all patches in <u>next</u> image
  - For each patch, compute covariance matrix $C^{Candidate}$
  - Find distance from C^{Model}

$$\rho(C^{Model}, C^{Candidate})$$

- Find patch region with <u>minimum</u> distance

Example



Results

Active Camera tracking
Feature space: [x y R G B]
Target: manually selected on 1st frame

Other

- Rotation invariance
  - Use radial distance instead of Cartesian position

    $f_k = [\ \mathbf{r(x,y)}\ \ R\ \ G\ \ B\ ]^T$
- Model Update
  - Calculate an “average”/smoothed covariance matrix by taking the mean of the models from the past few frames, then use that as C^{Model}
  - Must calculate mean on manifold (see paper)

11

Summary

- Algorithm for non-rigid object tracking
- Covariance Tracking
  - Covariance of spatial and statistical features
  - Low dimensional representation of object
  - Distances between matrices based on manifold distance

12