Object Tracking and Person Re-Identification on Manifolds

Conrad Sanderson

NICTA, Australia

- **e-mail:** conrad.sanderson [at] nicta.com.au
- **web:** http://conradsanderson.id.au

Presented at:

- Fatih Porikli, Mehrtash Harandi, Conrad Sanderson.
  Tutorial on Riemannian Geometry in Computer Vision.
  Asian Conference on Computer Vision (ACCV), 2014.

Part 1: Object Tracking on Manifolds

Published in:

- S. Shirazi, C. Sanderson, C. McCool, M. Harandi.
  Bags of Affine Subspaces for Robust Object Tracking.
  arXiv:1408.2313, 2014.
- Full paper: http://arxiv.org/pdf/1408.2313v2

Object tracking is hard:

- occlusions
- deformations
- variations in pose
- variations in scale
- variations in illumination
- imposters / similar objects

Tracking algorithms can be categorised into:

1. generative tracking
   - represent object through a particular appearance model
   - search for image area with most similar appearance
   - examples: mean shift tracker [1] and FragTrack [2]
2. discriminative tracking
   - treat tracking as binary classification task
   - discriminative classifier trained to explicitly separate object from non-object areas
   - example: Multiple Instance Learning (MILTrack) [3]
   - example: Tracking-Learning-Detection (TLD) [4]
   - requires larger training dataset than generative tracking

[1] Dorin Comaniciu et al.: *Kernel-based object tracking*. In: *IEEE PAMI* 25.5 (2003).

[2] A. Adam et al.: *Robust fragments-based tracking using the integral histogram*. In: *IEEE CVPR* (2006).

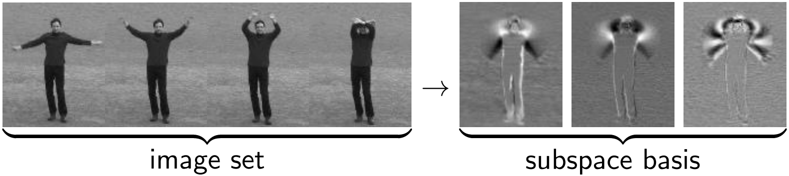
[3] B. Babenko et al.: *Robust object tracking with online multiple instance learning*. In: *IEEE PAMI* 33.8 (2011).

[4] Z. Kalal et al.: *Tracking-learning-detection*. In: *IEEE PAMI* 34.7 (2012).

Promising approach for generative tracking:

→ model object appearance via **subspaces**

- originated with the work of Black and Jepson [5]
- apply eigen decomposition on a set of object images
- resulting eigen vectors define a linear subspace
- subspaces able to capture perturbations of object appearance



[5] Michael J Black et al.: *EigenTracking: Robust matching and tracking of articulated objects using a view-based representation*. In: *IJCV* 26.1 (1998), pp. 63–84.

Many developments to address limitations:

- sequentially update the subspace [6][7]
- more robust update of the subspace [8][9][10]
- online updates using distances to subspaces on Grassmann manifolds [11]

But still not competitive with discriminative methods!

[6] Danijel Skocaj et al.: *Weighted and robust incremental method for subspace learning*. In: *ICCV* (2003).

[7] Yongmin Li: *On incremental and robust subspace learning*. In: *Pattern Recognition* 37.7 (2004).

[8] J. Ho et al.: *Visual tracking using learned linear subspaces*. In: *IEEE CVPR* (2004).

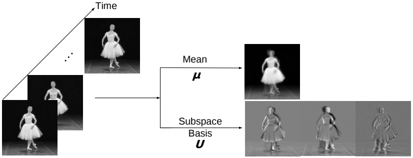
[9] Jongwoo Lim et al.: *Incremental learning for visual tracking*. In: *NIPS* (2004).

[10] D.A. Ross et al.: *Incremental learning for robust visual tracking*. In: *IJCV* 77.1-3 (2008).

[11] T. Wang et al.: *Online subspace learning on Grassmann manifold for moving object tracking in video*. In: *IEEE ICASSP* (2008).

Two major **shortcomings** in all subspace based trackers:

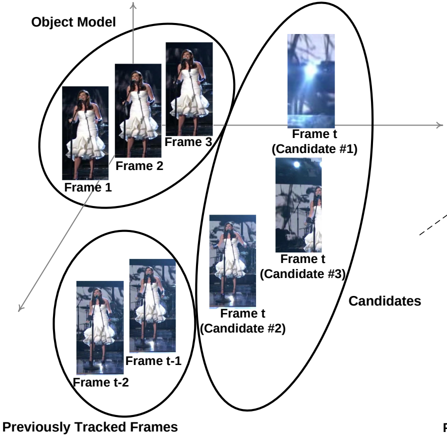
1. **mean** of the image set is not used
   - the mean can hold useful discriminatory information!

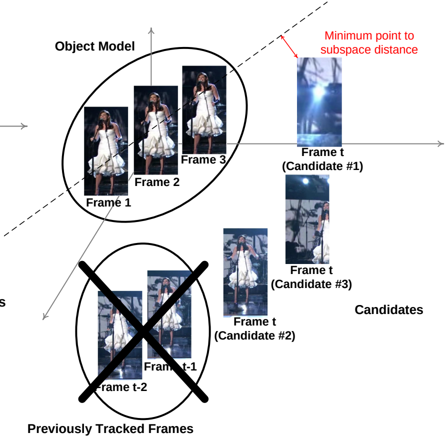


2. search for object location is typically done using **point-to-subspace distance**
   - compare a candidate image area from ONE frame against the model (multiple frames)
   - easily affected by drastic appearance changes (eg. occlusions)

Point-to-subspace distance

- each image is represented as a point
- object model (subspace) is conceptually represented as a line
- previously tracked frames are disregarded when comparing candidate frames to object model
- reduces memory of the system
- can easily lead to incorrect frame selection





Proposed Tracking Approach

Comprised of 4 intertwined components:

1. particle filtering framework (for efficient search)
2. model appearance of each particle as an **affine subspace**
   - takes into account tracking history (longer memory)
   - takes into account the mean
3. object model: **bag of affine subspaces**
   - continuously updated set of affine subspaces
   - longer memory
   - handles drastic appearance changes
4. likelihood of each particle according to object model:
   - (i) distance between means
   - (ii) distance between bases: **subspace-to-subspace distance**

1. Particle Filtering Framework

- Using standard particle filtering framework [12]
- History of object's location is parameterised as a distribution
  - set of particles represents the distribution
  - each particle represents a location and scale:

$$\boldsymbol{z}_i^{(t)} = [x_i^{(t)}, y_i^{(t)}, s_i^{(t)}]$$

- Use distribution to create a set of candidate object locations in a new frame
- Obtain **appearance** of each particle: $\mathcal{A}_i^{(t)}$
- Choose new location of object as the particle with highest likelihood according to **object model** $\mathcal{B}$:

$$\boldsymbol{z}_*^{(t)} = \boldsymbol{z}_j^{(t)}, \quad \text{where} \quad j = \underset{i}{\operatorname{argmax}} \; p\left(\mathcal{A}_i^{(t)} | \mathcal{B}\right)$$

[12] M.S. Arulampalam et al.: *A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking*. In: *IEEE Trans. Signal Processing* 50.2 (2002).

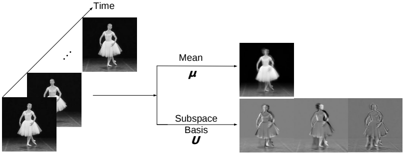
2. Model Appearance of Each Particle as an Affine Subspace

- Affine subspace represented as a 2-tuple:

$$\mathcal{A}_i^{(t)} = \left\{ \boldsymbol{\mu}_i^{(t)}, \boldsymbol{U}_i^{(t)} \right\}$$

$\boldsymbol{\mu}$: mean

$\boldsymbol{U}$: subspace basis



- Appearance includes:

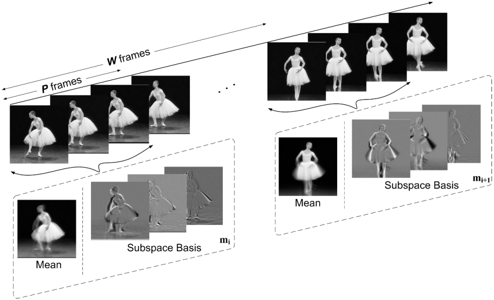
1. appearance of the *i*-th candidate location
2. appearance of tracked object in several preceding frames

3. Object Model: Bag of Affine Subspaces

- Drastic appearance changes (eg. occlusions) adversely affect subspaces
- Instead of modelling the object using only one subspace, use a **bag of subspaces**:

$$\mathcal{B} = \{\mathcal{A}_1, \cdots, \mathcal{A}_K\}$$

- Simple **model update:** the bag is updated every W frames by replacing the oldest affine subspace with the newest



4. Likelihood of Each Particle According to Object Model

- Particle filtering framework requires: $p\left(\mathcal{A}_i^{(t)}|\mathcal{B}\right)$
- Appearance of each candidate area: $\mathcal{A}_i^{(t)} = \left\{\boldsymbol{\mu}_i^{(t)}, \boldsymbol{U}_i^{(t)}\right\}$
- Object model: $\mathcal{B} = \{\mathcal{A}_1, \cdots, \mathcal{A}_K\}$
- Our definition: $p\left(\mathcal{A}_i^{(t)}|\mathcal{B}\right) = \sum_{k=1}^{K} \widehat{p}\left(\mathcal{A}_i^{(t)}|\mathcal{B}\left[k\right]\right)$
  - $\mathcal{B}\left[k\right]$ is the k-th affine subspace in bag $\mathcal{B}$
  - $\widehat{p}\left(\mathcal{A}_i^{(t)}|\mathcal{B}[k]\right) = \frac{p\left(\mathcal{A}_i^{(t)}|\mathcal{B}[k]\right)}{\sum_{j=1}^{N} p\left(\mathcal{A}_j^{(t)}|\mathcal{B}[k]\right)}$, where $N =$ num. of particles
  - $p\left(\mathcal{A}_i^{(t)}|\mathcal{B}[k]\right) \approx \exp\left\{-\underbrace{\mathrm{dist}(\mathcal{A}_i^{(t)}, \mathcal{B}[k])}_{\textbf{distance between affine subspaces}}\right\}$

- Define the **distance** between two affine subspaces as:

$$\text{dist}(\mathcal{A}_i, \mathcal{A}_j) = \alpha \, \widehat{\mathrm{d}}_o \left(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j\right) + (1-\alpha) \, \widehat{\mathrm{d}}_g \left(\boldsymbol{U}_i, \boldsymbol{U}_j\right)$$

  - $\widehat{\mathrm{d}}_o \left(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j\right) =$ normalised Euclidean distance between means
  - $\widehat{\mathrm{d}}_g \left(\boldsymbol{U}_i, \boldsymbol{U}_j\right) =$ normalised geodesic distance between bases

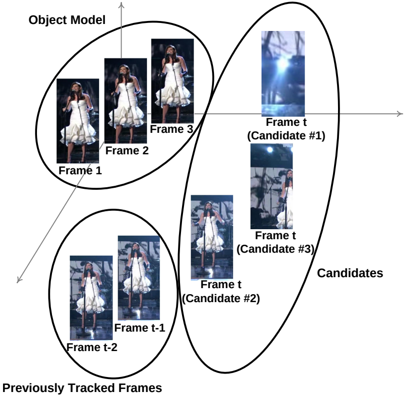
- Grassmann manifolds:
  - space of all n-dimensional linear subspaces of $\mathbb{R}^D$ for $0<n<D$
  - a point on Grassmann manifold $\mathcal{G}_{D,n}$ in a $D \times n$ matrix

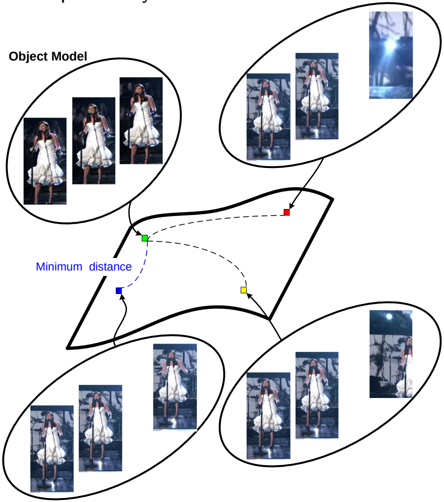
- Geodesic distance between subspaces $\boldsymbol{U}_i$ and $\boldsymbol{U}_j$ is:

$$\mathrm{d}_g \left(\boldsymbol{U}_i, \boldsymbol{U}_j\right) = \left\| [\theta_1, \theta_2, \cdots, \theta_n] \right\|$$

  - $[\theta_1, \theta_2, \cdots, \theta_n] =$ vector of principal angles
  - $\theta_1 =$ smallest angle btwn. all pairs of unit vectors in $\boldsymbol{U}_i$ and $\boldsymbol{U}_j$
  - principal angles are computed via SVD of $\boldsymbol{U}_i^T \boldsymbol{U}_j$

- each image set is represented as a point on a Grassmann manifold
- explicitly takes into account previously tracked frames





Computational Complexity

- Generation of new affine subspace:
  - patch size: $H_1 \times H_2$
  - represent patch as vector: $D = H_1 \times H_2$
  - use patches from P frames
  - $\therefore$ SVD of $D \times P$ matrix
  - $D >> P$
  - using optimised thin SVD[13]: $\mathcal{O}\left(Dn^2\right)$ operations
  - $n =$ number of basis vectors
- To keep computational requirements relatively low:
  - patch size: 32×32
  - number of frames: 5
  - number of basis vectors: 3

[13] Matthew Brand: *Fast low-rank modifications of the thin singular value decomposition*. In: *Linear Algebra and its Applications* 415.1 (2006).

Comparative Evaluation

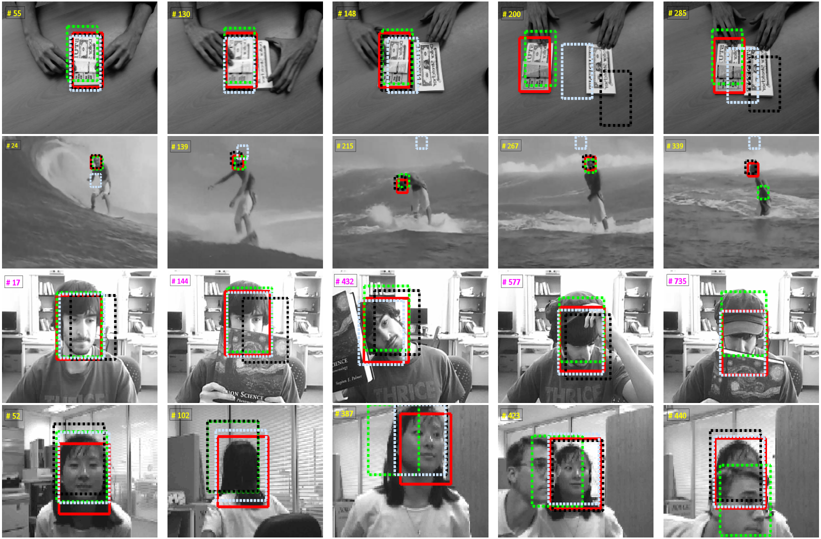
- Evaluation on 8 commonly used videos in the literature
- Compared against recent tracking algorithms:
  - Tracking-Learning-Detection (TLD)[14]
  - Multiple Instance Learning (MILTrack) [15]
  - Sparse Collaborative Model (SCM) [16]
- Qualitative and quantitative evaluation

[14] Z. Kalal et al.: *Tracking-learning-detection*. In: *IEEE PAMI* 34.7 (2012).

[15] B. Babenko et al.: *Robust object tracking with online multiple instance learning*. In: *IEEE PAMI* 33.8 (2011).

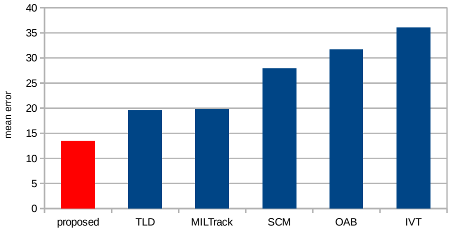
[16] Wei Zhong et al.: *Robust object tracking via sparsity-based collaborative model*. In: *IEEE CVPR* (2012).

proposed method	TLD (PAMI 2012)	MILTrack (PAMI 2011)	SCM (CVPR 2012)



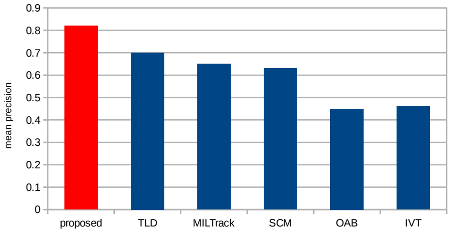
Quantitative Results

- Used two measures:
  1. **centre location error:** distance between the centre of the bounding box and the ground truth object position
  2. **precision:** percentage of frames where the estimated object location is within a pre-defined distance to ground truth



average error

(lower = better)



average precision

(higher = better)

Future Work

- Affected by motion blurring (rapid motion or pose variations)
- Better update scheme by measuring the effectiveness of new affine subspace before adding it to the bag
- Allow bag size and update rate to be dynamic, possibly dependent on tracking difficulty

Part 2: Person Re-Identification on Manifolds

Published in:

- A. Alavi, Y. Yang, M. Harandi, C. Sanderson.
  Multi-Shot Person Re-Identification via Relational Stein Divergence.
  IEEE International Conference on Image Processing (ICIP), 2013.
- official version: http://dx.doi.org/10.1109/ICIP.2013.6738731
- arXiv pre-print: http://arxiv.org/pdf/1403.0699v1

- Given images of a person from camera view 1, find matching person from camera view 2
- Difficult:
  - imperfect person detection / localisation
  - large pose changes
  - occlusions
  - illumination changes
  - low resolution

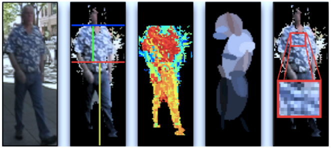
Popular Previous Approaches

Partial Least Squares (PLS) based [17]

- decompose an image into overlapping blocks
- extracts features from each block: textures, edges, colours
- concatenated into one feature vector (high dimensional)
- learn discriminative dimensionality reduction for each person
- classification: projection to each model + Euclidean distance
- **downsides:**
  - concatenation = fixed spatial relations between blocks
  - ∴ **does not allow for movement of blocks**!
  - ∴ **easily affected** by imperfect localisation and pose variations

[17] W.R. Schwartz et al.: *Learning discriminative appearance-based models using partial least squares*. In: *SIBGRAPI* (2009).

Symmetry-Driven Accumulation of Local Features (SDALF)[18]



- foreground detection
- two horizontal axes of asymmetry to isolate: head, torso, legs
- use vertical axes of appearance symmetry for torso and legs
- extract: HSV histogram, stable colour regions, textures
- estimation of symmetry affected by deformations & pose variations:
  - ∴ **noisy features**

[18] M. Farenzena et al.: *Person re-identification by symmetry-driven accumulation of local features*. In: *CVPR* (2010).

Proposed Method

- Aim to obtain a compact & robust representation of an image:
  - allow for imprecise person detection
  - allow for deformations
  - $\therefore$ do not use rigid spatial relations
  - do not use brittle feature extraction based on symmetry
- Steps:
  1. foreground estimation
  2. for each foreground pixel, extract feature vector containing colour and local texture information
  3. represent the set of feature vectors as a covariance matrix
  4. covariance matrix is a point on a Riemannian manifold
  5. map matrix from R. manifold to vector in Euclidean space, **while taking into account curvature of the manifold!**
  6. use standard machine learning for classification

Feature Extraction

- For each foreground pixel, extract feature vector:

$$\boldsymbol{f} = [\ x,\ y,\ HSV_{xy},\ \Lambda_{xy},\ \Theta_{xy}\ ]^T$$

where

  - $HSV_{xy} = [H_{xy}, S_{xy}, \widehat{V}_{xy}]$ = colour values of the HSV channels
  - $\Lambda_{xy} = [\lambda^R_{xy}, \lambda^G_{xy}, \lambda^B_{xy}]$ = gradient magnitudes
  - $\Theta_{xy} = [\theta^R_{xy}, \theta^G_{xy}, \theta^B_{xy}]$ = gradient orientations

- (not limited to above, can certainly use other features)
- Given set $F = \{\boldsymbol{f}_i\}_{i=1}^N$, calculate covariance matrix:

$$\boldsymbol{C} = \frac{1}{N-1} \sum_{i=1}^N (\boldsymbol{f}_i - \boldsymbol{\mu})(\boldsymbol{f}_i - \boldsymbol{\mu})^T$$

  - low dimensional representation, independent of image size

How to Compare Covariance Matrices?

- Naive method:
  - brute-force vectorisation of matrix
  - use Euclidean distance between resultant vectors

- Naive method kind-of works, BUT:
  - covariance matrix = symmetric positive definite (SPD) matrix
  - space of SPD matrices = interior of a convex cone in $\mathbb{R}^{D^2}$
  - space of SPD matrices = Riemannian manifold[19]
  - $\therefore$ covariance matrix = point on a Riemannian manifold
  - naive method **disregards** curvature of manifold!
  - geodesic distance: shortest path along the manifold (eg. on a sphere)

---

[19] X. Pennec et al.: *A Riemannian Framework for Tensor Computing*. In: *IJCV* 66.1 (2006).

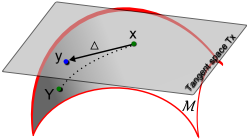
How to Measure Distances on Riemannian Manifolds?

- Use Affine Invariant Riemannian Metric (AIRM) [20]:

$$\delta_R(\boldsymbol{A}, \boldsymbol{B}) = \left\| \log\left(\boldsymbol{B}^{-\frac{1}{2}} \boldsymbol{A} \boldsymbol{B}^{-\frac{1}{2}} \right) \right\|_F$$

  - intensive use of matrix inverses, square roots, logarithms [21]
  - $\therefore$ **computationally demanding!**

- Choose a tangent pole, and map all points to tangent space



  - tangent space is Euclidean space
  - faster, but less precise
  - **true geodesic distances are only to the tangent pole!**

[20] X. Pennec et al.: *A Riemannian Framework for Tensor Computing*. In: *IJCV* 66.1 (2006).

[21] V. Arsigny et al.: *Log-Euclidean metrics for fast and simple calculus on diffusion tensors*. In: *Magnetic Resonance in Medicine* 56.2 (2006).

Stein Divergence

- Related to AIRM, but much faster [22]

$$\delta_S(\boldsymbol{A}, \boldsymbol{B}) = \log\left(\det\left(\frac{\boldsymbol{A}+\boldsymbol{B}}{2}\right)\right) - \frac{1}{2}\log\left(\det\left(\boldsymbol{AB}\right)\right)$$

- divergence, **not a true distance!**

Proposed: Relational Divergence Classification

- Obtain a set of training covariance matrices $\{\boldsymbol{T}\}_{i=1}^{N}$
- For matrix $\boldsymbol{C}$, calculate its Stein divergence to each training covariance matrix:

$$[\ \delta_S(\boldsymbol{C}, \boldsymbol{T}_1)\ \ \delta_S(\boldsymbol{C}, \boldsymbol{T}_2)\ \ \cdots\ \ \delta_S(\boldsymbol{C}, \boldsymbol{T}_N)\ ]\ \in\ \mathbb{R}^N$$

- In effect, we have **mapped** matrix $\boldsymbol{C}$ from manifold space to Euclidean space, while taking into account manifold curvature
- Can now use **standard** machine learning methods

[22] S. Sra: *A new metric on the manifold of kernel matrices with application to matrix geometric means*. In: *NIPS* (2012).

Comparative Evaluation

- After mapping from manifold space to Euclidean space, use LDA based classifier
- Use ETHZ dataset [23]
  - captured from a moving camera
  - occlusions and wide variations in appearance
- Compare with:
  - directly using the Stein divergence
  - Histogram Plus Epitome (HPE) [24]
  - Partial Least Squares (PLS)[25]
  - Symmetry-Driven Accumulation of Local Features (SDALF)[26]

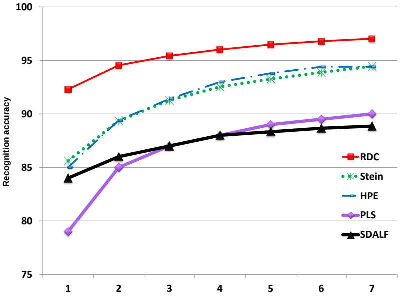
[23] A. Ess et al.: *Depth and Appearance for Mobile Scene Analysis*. In: *ICCV* (2007).

[24] Loris Bazzani et al.: *Multiple-Shot Person Re-identification by HPE Signature*. In: *ICPR* (2010).

[25] W.R. Schwartz et al.: *Learning discriminative appearance-based models using partial least squares*. In: *SIBGRAPI* (2009).

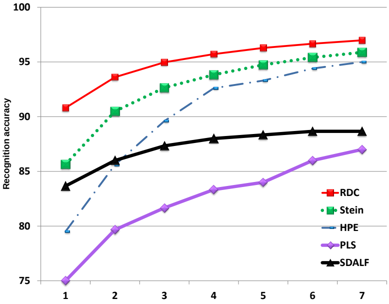
[26] M. Farenzena et al.: *Person re-identification by symmetry-driven accumulation of local features*. In: *CVPR* (2010).

ETHZ sequence 1



rank

ETHZ sequence 2



rank

RDC	=	Relational Divergence Classification (proposed method)
Stein	=	direct use of Stein divergence (no mapping)
HPE	=	Histogram Plus Epitome
PLS	=	Partial Least Squares
SDALF	=	Symmetry-Driven Accumulation of Local Features

- **Questions?**
  e-mail: conrad.sanderson [at] nicta.com.au

- More papers on machine learning & computer vision using manifolds:
  http://conradsanderson.id.au/papers.html