The chains model for detecting parts by their context

Leonid Karlinsky, Michael Dinerstein, Daniel Harari, and Shimon Ullman

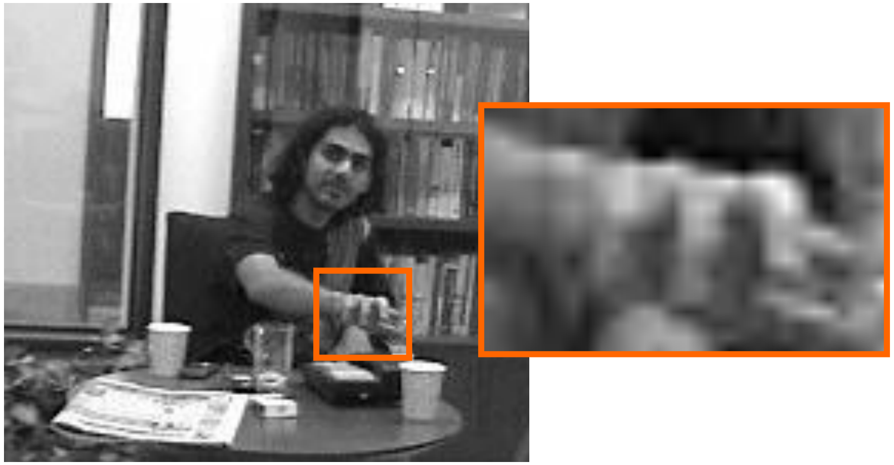
CVPR
2010 - San Francisco

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

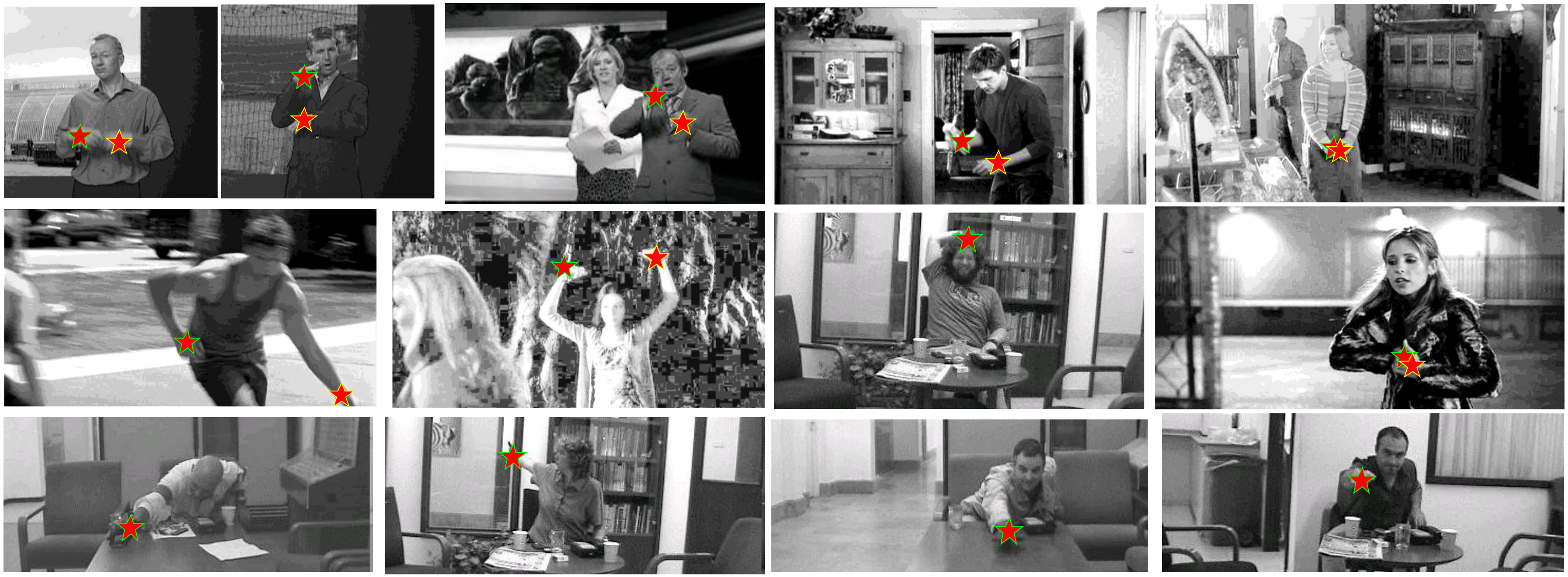
Goals & Intuition

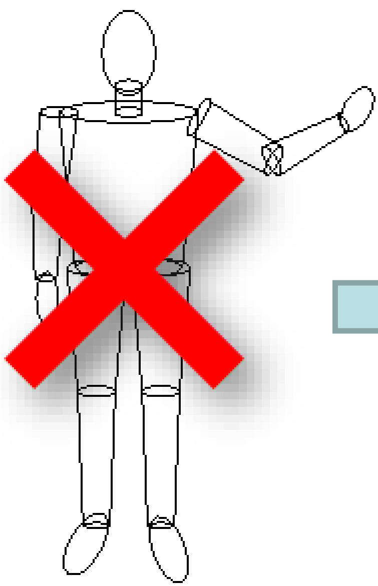
- Detects a “difficult” part by extending from an “easy” reference part
- Detect by marginalizing over chains of intermediate features

Use the face to disambiguate the hand



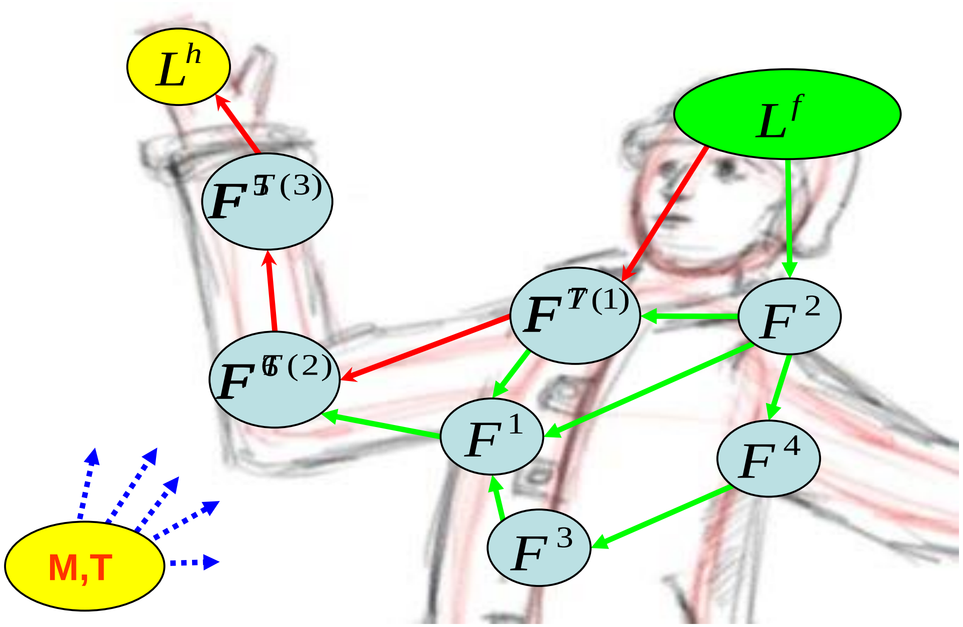
Examples & Generality





Same approach for any parts of any object!

The chains model



Chains model

Exact inference:

Pre-processing:

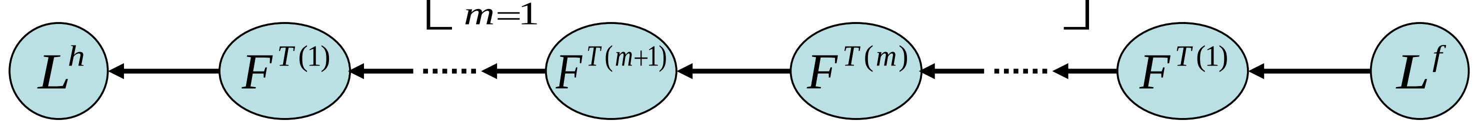
- Done over a simple chains graph
- Detect the face
- Extract features

Approximate inference:

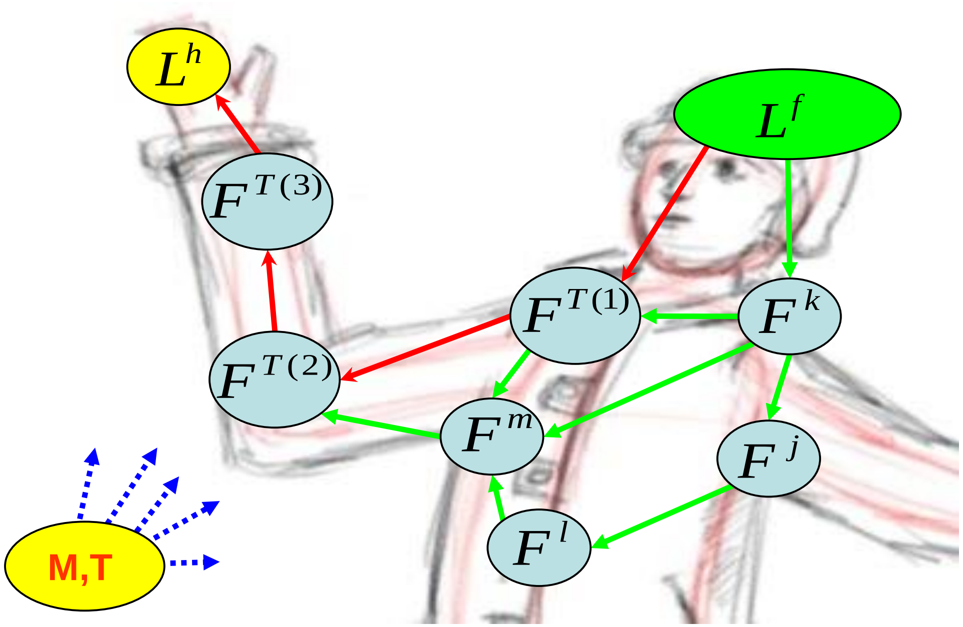
- Sum over all chains (limit the max. length)
- Model the probability that hand is in L^h
- Can be easily computed by matrix multiplication
- Co-occurrence of features

$$P_{CH}\left(M,T,L^h,L^f,\left\{F^i\right\}\right)=P(M,T)\cdot P\left(L^f\right)\cdot \boxed{\prod_{i\notin T} P_{obj+bg}\left(F^i\right)\cdot}$$

$$\boxed{P_{obj}\left(L^h \middle| F^{T(M)}\right)\cdot\left[\prod_{m=1}^{M-1} P_{obj}\left(F^{T(m+1)} \middle| F^{T(m)}\right)\right]\cdot P_{obj}\left(F^{T(1)} \middle| L^f\right)}$$



Inference



Chains model

$$\sum_{M,T} P_{CH}\left(M,T,L^h,L^f,\{F^i\}\right) \approx \underbrace{\left[\dots d_i \dots\right]}_{D} \cdot \underbrace{\sum_M \begin{bmatrix} & \dots & \\ \dots & a_{ij} & \dots \\ & \dots & \end{bmatrix}^M}_{C} \cdot \underbrace{\begin{bmatrix}\dots\\ s_j \\ \dots\end{bmatrix}}_{S}$$

voting probabilities = $D = \left\langle d_i = P_{obj}\left(L^h \middle| F^i\right)\right\rangle$

Entry i-j is the marginal over all chains between F^i and F^j = C

object vs. background likelihood ratios = $S = \left\langle s_j = \dfrac{P_{obj}\left(F^j \middle| L^f\right)}{P_{obj+bg}\left(F^j\right)}\right\rangle$

Training & probability estimation

Training:

- Get example images with detected reference part and target part marked by single point.

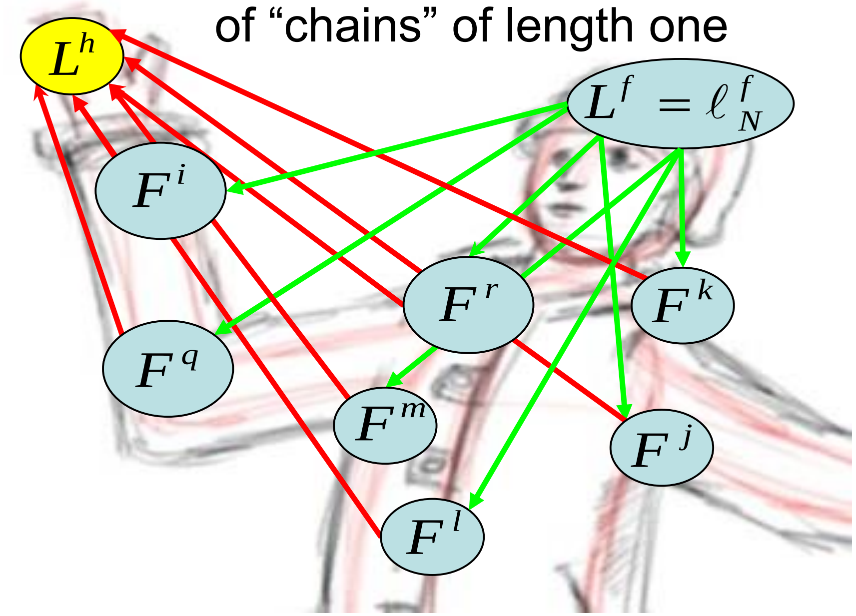
Non-parametric probability estimation in a nutshell:

- Store training image features and pairs of nearby features* in an efficient ANN search structures (KD-trees)
- Compute the estimated probability by approximate KDE
- E.g.: $P_{obj}\left(F^i, F^j\right)$ - search for K nearest neighbors of $\left[SIFT^i, SIFT^j, x^i - x^j\right]$, compute $P_{obj}\left(F^i, F^j\right) \approx \sum_{k=1}^{K} \exp\left(-\frac{d_k^2}{2\sigma^2}\right)$

* use a feature selection strategy (see paper) to reduce the number of features and pairs

Chains & Stars

Star model – a special case of “chains” of length one



Limitations of the star model:

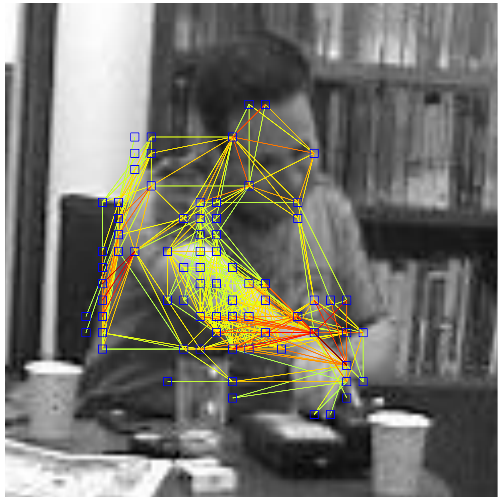
1. Far away features do not contribute.
2. Features contribute to detection without testing for connectivity with the reference part.

$$P_{STAR}\left(L^h, L^f, \{F_i\}\right) = P\left(L^h\right) \cdot P\left(L^f\right) \cdot \prod_i \left[\alpha \cdot P_{obj}\left(F_i \middle| L^h, L^f\right) + (1-\alpha) \cdot P_{obj+bg}\left(F_i \middle| L^f\right)\right]$$

star result | chains result | chains graph

Feature graph examples

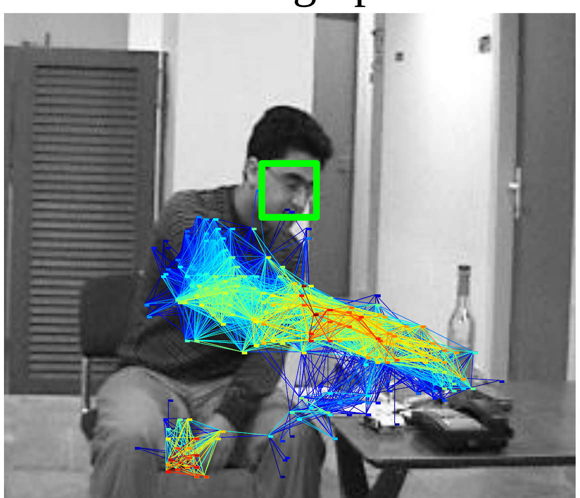
chains graph



chains result

---

chains graph



chains result

Results

Buehler et al 2008 – sign language dataset

Training:
600 frames of a news sequence

Testing:
195 frames from 5 other sequences

Our result:
84.9% detection

Best in the literature:
79% **[Kumar et al, 2009]**
(avg. over 6 body parts)

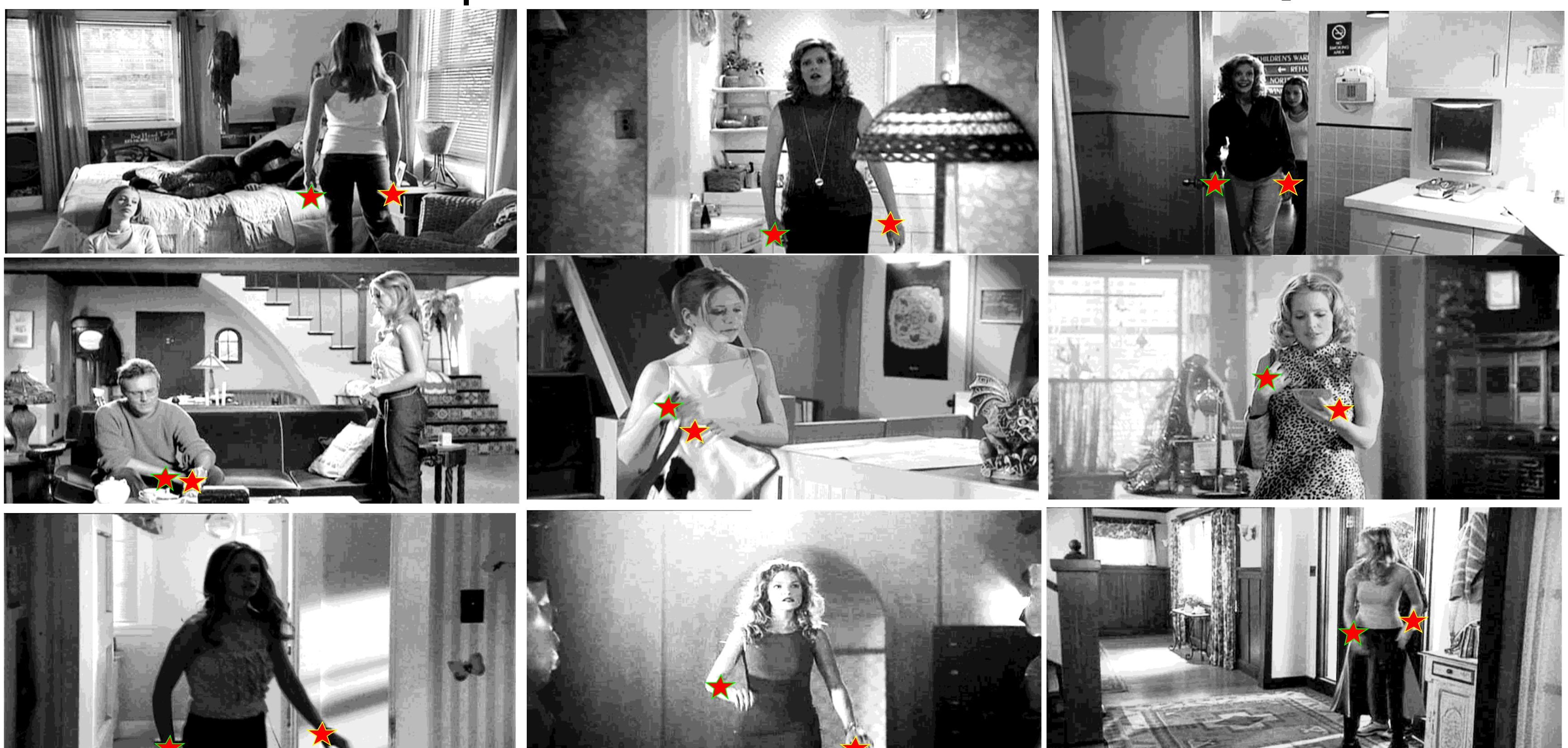
Ferrari et al 2008 - Buffy dataset

Training:
movies shot
in our lab

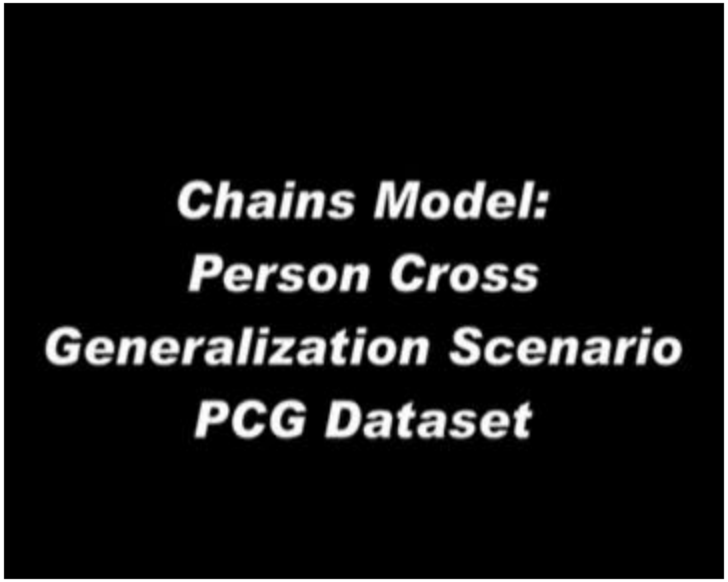
Testing:
308 frames
/ 3 episodes

Our result:
47% detection

Best in the literature:
46.1% **[Andriluka et al, 2009]**



Person cross generalization scenario (86.9%)



Self trained scenario (87%)

Full generalization scenario (73.2%)

Buehler et al. 2008 – BBC news dataset (96.8%)

Best in the literature: 95.6% [Buehler et al, 2008]

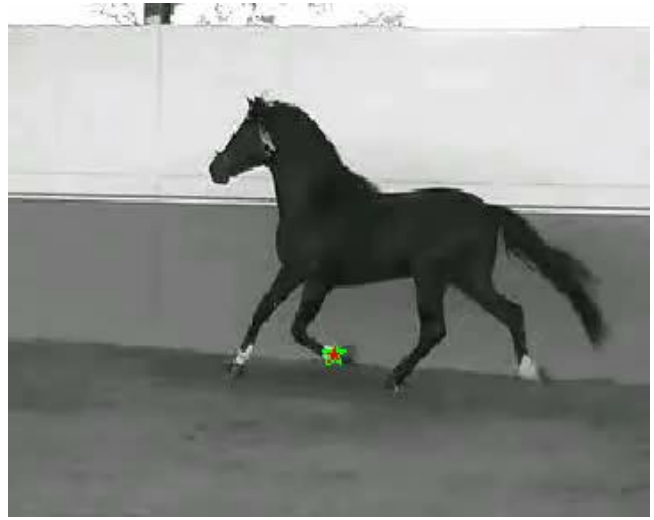
UIUC cars dataset

Training:
550 positive /
500 neg.

Testing:
200 cars in
170 images

Our result:
99.5% detection

Best in the literature:
99.9% [Mutch & Lowe, 2006]



Same approach (same params) works for other object types (out of the box)

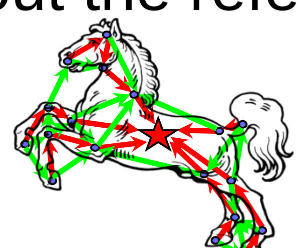
Our result:

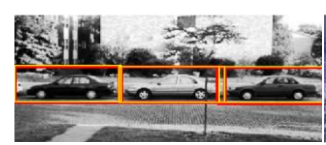
63.1% - 1st max only

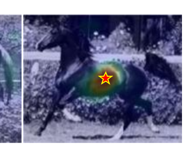
90.3% - 1st & 2nd max

Summary:

- A single non-parametric model for any object
- Detect difficult parts extending from the ‘easy’ parts
- Near features vote directly, far features vote through chains
- Can be used without the reference part and for full objects







detailed numerical results…

The End

	Chains				Star			
Scenario + Dataset	1 max	2 max	3 max	4 max	1 max	2 max	3 max	4 max
self trained (ST)	87	93.9	95.2	95.9	46.9	65	75.7	81.2
self trained (Horse)	63.1	90.3	93.4	94.6	50.2	61.6	68	74
person cross-generalization (PCG)	86.9	93.5	95.4	96.1	57.9	74.1	81.9	85.5
full generalization (ST)	73.2	86.6	90	92.2	37.6	52	66.9	74.7

	Chains				[Buehler et al., 2008]	[Kumar et al., 2009]	[Ferrari et al., 2008]	[Andriluka et al., 2009]
Setting	1 max	2 max	3 max	4 max	1 max	1 max	1 max	1 max
BBC news	96.8	99	99.5	99.6	95.6	86.37[1]	-	-
5 signers	84.9	92.8	95.4	96.7	-	78.95[1]	-	-
Buffy	47	53.9	56.7	59.3	-	39.2[1]	56[1]	46.1

1 – average on 6 upper body parts
all except Andriluka et al. and us use tracking
all except us use color

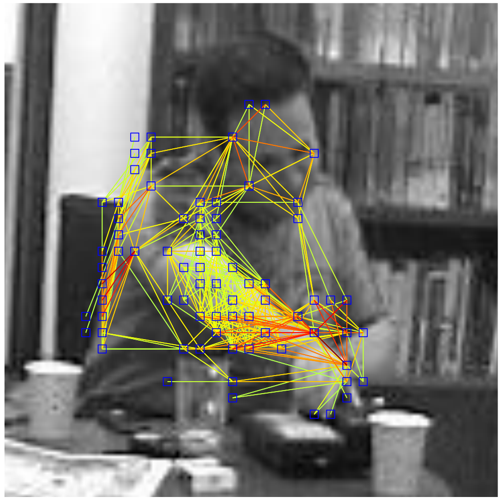
Methods	**Precision-Recall at EER - cars**
Mutch and Lowe, 2006	99.9%
Leibe et al, 2008	97.5%
Lampert et al, 2008	98.5%
Gall and Lempitsky, 2009	98.5%
chains model (trained on UIUC)	*99.5%*
chains model (trained on Caltech-101 cars)	*93.4%*

Methods	**Precision-Recall at EER - horses**
Shotton et al, 2008	95.7%
Gall and Lempitsky, 2009	93.9%
chains model	*97.5±1%*

back…

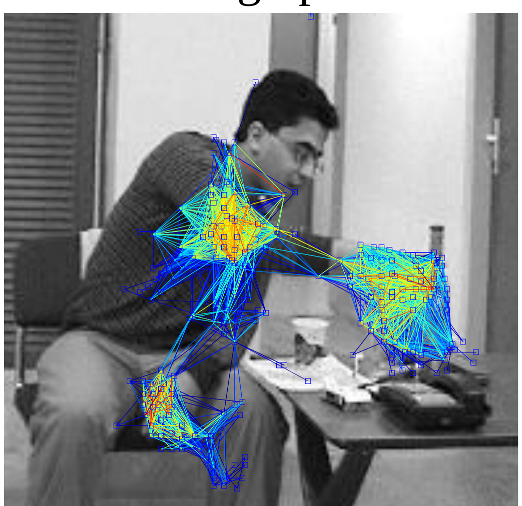
Feature graph examples

chains graph*



chains result

chains graph*



chains result

* weights – shown in logarithmic color scale