Beyond Covariance: Higher-order Tensor Descriptors and Applications in Computer Vision

P. Koniusz

National ICT Australia (NICTA)

In collaboration with:
A. Cherian, F. Porikli, P. H. Gosselin, F. Yan, K, Mikolajczyk.

piotr.koniusz@nicta.com.au

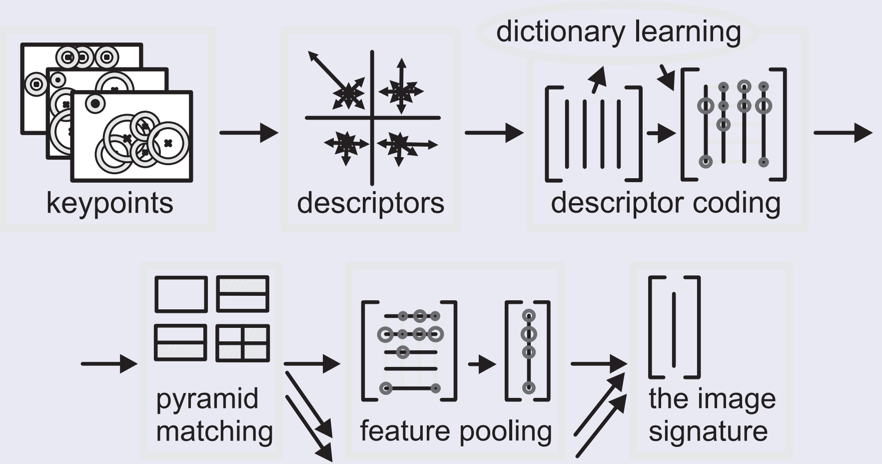
July 26, 2016

The Content

- Recap of Bag-of-Words (universal baseline to study).
- Second- and higher-order occurrence pooling.
- Intuitive explanation: uncertainty in max-pooling.
- Evaluations and comparisons to fisher vector encoding.
- Region covariance descriptors.
- Embedding into RKHS+Third-order Super-symmetric Tensor descriptors.
- Details of linearization process.
- Sparse coding for TOSST descriptor.
- Evaluations of TOSST on texture recognition.
- Action recognition from 3D skeletons: Sequence and Dynamics Compatibility Kernels.
- Details of eigenvalue Power Normalisation.
- Results on 3D skeleton action classification.
- Natural Inner Product on Gaussians and Deep Architectures.

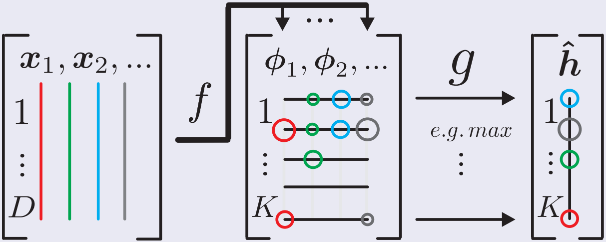
Introduction

Bag-of-Words (First-order Approaches)



First-order Occurrence Pooling

- The local descriptors $\boldsymbol{x}$ are extracted from an image and coded by f that operates on columns.
- Pooling g aggregates visual words from the mid-level features ϕ along rows:



- Let me remind the following three steps (without Pyramid Matching):

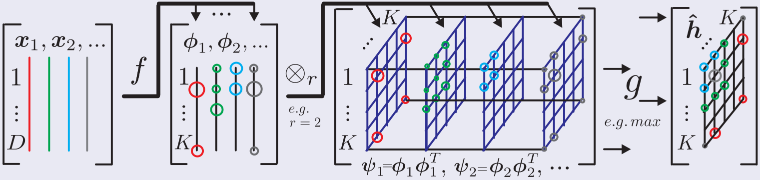
$$\phi_n = f(\boldsymbol{x}_n, \mathbf{D}), \quad \forall n \in \mathcal{N} \qquad \text{(encode)} \qquad (1)$$

$$\hat{h}_k = g\left(\{\phi_{kn}\}_{n\in\mathcal{N}}\right) \qquad \text{(pool)} \qquad (2)$$

$$\boldsymbol{h} = \hat{\boldsymbol{h}}/\|\hat{\boldsymbol{h}}\|_2 \qquad \text{(normalise)} \qquad (3)$$

Higher-order Occurrence Pooling

- Note that Fisher Vector Encoding and Vector of Locally Aggregated Tensors use the second-order statistics and Power Normalisation.
- BoW can employ the second-order statistics with $\uparrow\otimes_r$, e.g. $\uparrow\otimes_2\phi = \phi\phi^T$:



- Formally, this can be expressed in four steps:

$$\phi_n = f(\boldsymbol{x}_n, \mathbf{D}), \quad \forall n \in \mathcal{N} \qquad \text{(encode)} \tag{4}$$

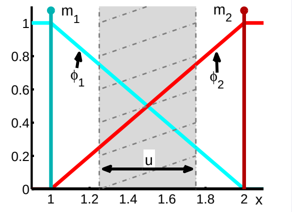
$$\psi_n = \otimes_r \phi_n \qquad \text{(co-occurrences)} \tag{5}$$

$$\psi_n := u_:(\psi_n) \qquad \text{(vectorise)} \tag{6}$$

$$\hat{h}_k = g\left(\{\psi_{kn}\}_{n\in\mathcal{N}}\right) \qquad \text{(pool)} \tag{7}$$

$$\boldsymbol{h} = \hat{\boldsymbol{h}} / \|\hat{\boldsymbol{h}}\|_2 \qquad \text{(normalise)} \tag{8}$$

Uncertainty in max-pooling



- Two linear slopes (LLC) - coding values ϕ_1 and ϕ_2 for any $1 \leq x \leq 2$.
- Draw randomly descriptors from this interval and apply max-pooling.
- If we were to draw several times (a) $x = 1.5$, we would obtain:

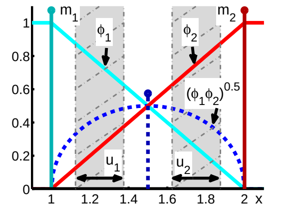
	ϕ_1	ϕ_2
	0.5	0.5
	...	
	0.5	0.5
max	0.5	0.5

For (b) $x_1 = 1$, $x_2 = 2$, $x_3 = 1.5$, we get:

	ϕ_1	ϕ_2
	0	1
	1	0
	0.5	0.5
max	1	1

- Position of the descriptor $x = 1.5$ in (a) can be uniquely retrieved from ϕ because $f^{-1}\left([0.5, 0.5]^T\right) = 1.5$.
- Position of $x_3 = 1.5$ in (b) is lost as $f^{-1}\left([1, 1]^T\right) \in [1; 2]$.

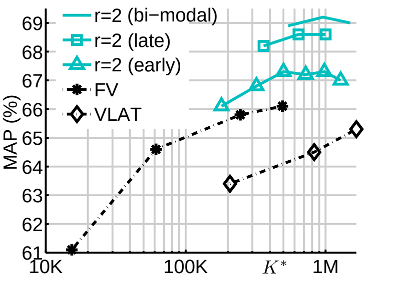
Uncertainty in max-pooling



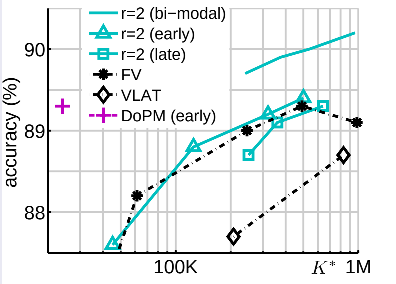
- Two linear slopes (LLC) display coding values ϕ_1 and ϕ_2 for any $1 \leq x \leq 2$.
- Co-occurrences improve on the making effect, e.g. take co-occurrence $\phi_1\phi_2$.
- It results in a new maximum for $x = 1.5$ - masking region u is now split in two smaller regions u_1 and u_2.

Second-order Occurrence Pooling: results

PascalVOC07, Flower102, Opponent SIFT, Sparse Coding



(*a*) VOC07

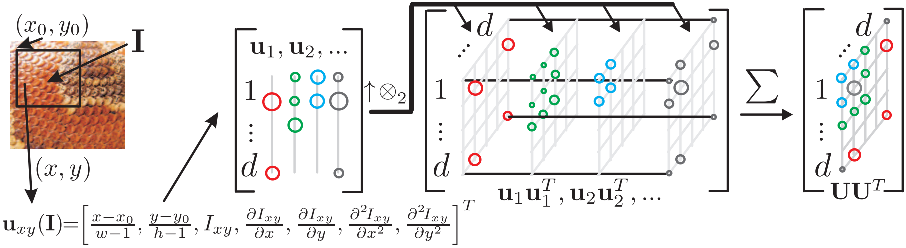


(*b*) Flower102

- K^* - length of the image signature.
- $r\!=\!2$ - Second-order Occurrence Pooling
  FV/VLAT - Fisher Kernels/Vector of Locally Aggregated Tensors.
- Fusions: early - descriptor level, late - kernel level, bi-modal - tensor.
- DoPM - a first-order method.

TOSST Texture Descriptors

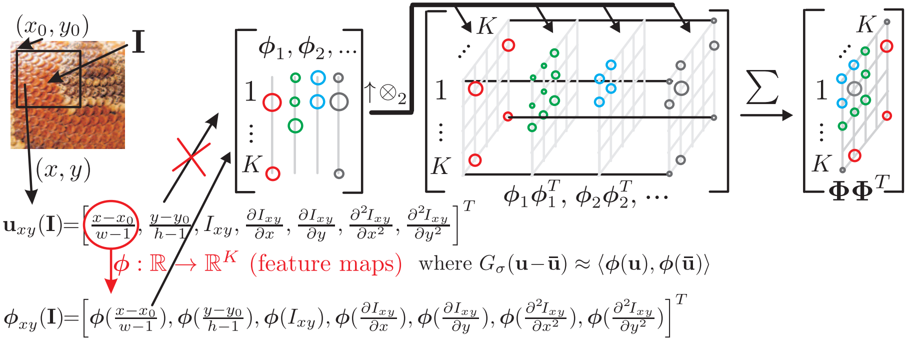
- Region covariance descriptors (co-occurrences).
- They typically use outer-product of low-level feature vectors $\uparrow\otimes_2\mathbf{u} = \mathbf{u}\mathbf{u}^T$.



- Low-level features $\mathbf{u}$ (linear). Non-linear features are better. We embed $\mathbf{u}$ into RKHS/linearise the RBF kernel by feature maps.

TOSST Texture Descriptors

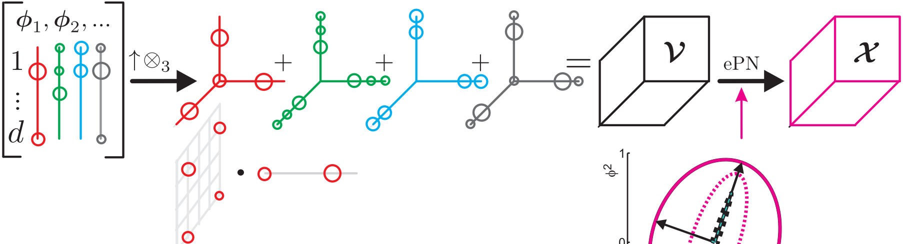
- Non-linear co-occurrence descriptors.



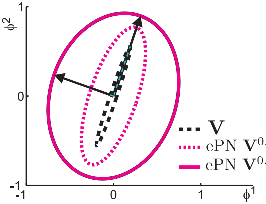
- Can we form more informative co-occurrences?
  Yes, we extend $\uparrow\otimes_2$ to third-order outer product $\uparrow\otimes_3$.

TOSST Texture Descriptors

- Non-linear third-order descriptors
  +eigenvalue Power Normalization (ePN).



- Eigenvalue Power Normalisation prevents correlated signal bursts. Imagine the largest eigenvalue repesents the count of pattern of brick and the 2nd represents the tree bark. Surely, the amount of brick and bark patterns should not affect the prediction. But it does!



- Higher-order models can be derived analytically.

Higher-order Occurrence Pooling: derivation

- Assume a kernel, e.g. RBF and its linearisation given by: $ker\left(\mathbf{u},\bar{\mathbf{u}}\right)\approx\left\langle\phi,\bar{\phi}\right\rangle$.
- Assume the dot product $\left\langle\phi,\bar{\phi}\right\rangle$ on a pair of features and polynomial kernel: $\left\langle\phi,\bar{\phi}\right\rangle^r$, $r\geq 2$.
- Define a sum kernel between two sets of features $\boldsymbol{U}=\{\mathbf{u}_n\}_{n\in\mathcal{N}}$ and $\bar{\boldsymbol{U}}=\{\bar{\mathbf{u}}_{\bar{n}}\}_{\bar{n}\in\bar{\mathcal{N}}}$ for two images/regions/sequences (anything you like):

$$
\begin{aligned}
Ker\left(\boldsymbol{U},\bar{\boldsymbol{U}}\right) &= \frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}}\frac{1}{|\bar{\mathcal{N}}|}\sum_{\bar{n}\in\bar{\mathcal{N}}} ker\left(\mathbf{u}_n,\bar{\mathbf{u}}_{\bar{n}}\right)^r \\
&\approx \frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}}\frac{1}{|\bar{\mathcal{N}}|}\sum_{\bar{n}\in\bar{\mathcal{N}}} \left\langle\phi_n,\bar{\phi}_{\bar{n}}\right\rangle^r \\
&= \frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}}\frac{1}{|\bar{\mathcal{N}}|}\sum_{\bar{n}\in\bar{\mathcal{N}}} \left(\sum_{k=1}^{K}\phi_{kn}\bar{\phi}_{k\bar{n}}\right)^r \qquad (9)
\end{aligned}
$$

Higher-order Occurrence Pooling: derivation

- The rightmost summation can be re-expressed as a dot product of two outer-products of order r on ϕ:

$$\left(\sum_{k=1}^{K}\phi_{kn}\bar{\phi}_{k\bar{n}}\right)^{r}=\sum_{k^{(1)}=1}^{K}\dots\sum_{k^{(r)}=1}^{K}\phi_{k^{(1)}}\bar{\phi}_{k^{(1)}}\cdot\ldots\cdot\phi_{k^{(r)}}\bar{\phi}_{k^{(r)}}=\left\langle\otimes_r\boldsymbol{\phi}_n,\otimes_r\bar{\boldsymbol{\phi}}_{\bar{n}}\right\rangle_F \tag{10}$$

- Now, the problem is further simplified:

$$Ker\left(\boldsymbol{U},\bar{\boldsymbol{U}}\right)\approx Ker'\left(\boldsymbol{\Phi},\bar{\boldsymbol{\Phi}}\right)=\frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}}\frac{1}{|\bar{\mathcal{N}}|}\sum_{\bar{n}\in\bar{\mathcal{N}}}\left\langle\otimes_r\boldsymbol{\phi}_n,\otimes_r\bar{\boldsymbol{\phi}}_{\bar{n}}\right\rangle_F$$

$$=\left\langle\frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}}\otimes_r\boldsymbol{\phi}_n,\frac{1}{|\bar{\mathcal{N}}|}\sum_{\bar{n}\in\bar{\mathcal{N}}}\otimes_r\bar{\boldsymbol{\phi}}_{\bar{n}}\right\rangle_F=\left\langle\underset{n\in\mathcal{N}}{\text{Avg}}\left(\otimes_r\,\boldsymbol{\phi}_n\right),\underset{\bar{n}\in\bar{\mathcal{N}}}{\text{Avg}}\left(\otimes_r\bar{\boldsymbol{\phi}}_{\bar{n}}\right)\right\rangle_F$$

- We introduce operator $\boldsymbol{\mathcal{G}}$ (similarity for matrices/tensors, e.g. ePN):

$$Ker^*\left(\boldsymbol{\Phi},\bar{\boldsymbol{\Phi}}\right)=\left\langle\boldsymbol{\mathcal{G}}\left(\frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}}\otimes_r\boldsymbol{\phi}_n\right),\boldsymbol{\mathcal{G}}\left(\frac{1}{|\bar{\mathcal{N}}|}\sum_{\bar{n}\in\bar{\mathcal{N}}}\otimes_r\bar{\boldsymbol{\phi}}_{\bar{n}}\right)\right\rangle_F \tag{11}$$

Sparse Coding for Third-order Tensor Descriptors (TSC)

- However, $\boldsymbol{\mathcal{X}}$ is cubic w.r.t. size of features. We propose Sparse Coding for Third-order Tensor Descriptors.
- We can learn a dictionary to encode TOSST:

$$\underset{\substack{\boldsymbol{\mathcal{B}}_1,\ldots,\boldsymbol{\mathcal{B}}_K\\ \boldsymbol{\alpha}^1,\ldots,\boldsymbol{\alpha}^N}}{\arg\min} \sum_{n=1}^{N} \left\| \boldsymbol{\mathcal{X}}_n - \sum_{k=1}^{K} \boldsymbol{\mathcal{B}}_k \alpha_k^n \right\|_F^2 + \lambda \left\|\boldsymbol{\alpha}^n\right\|_1. \tag{12}$$

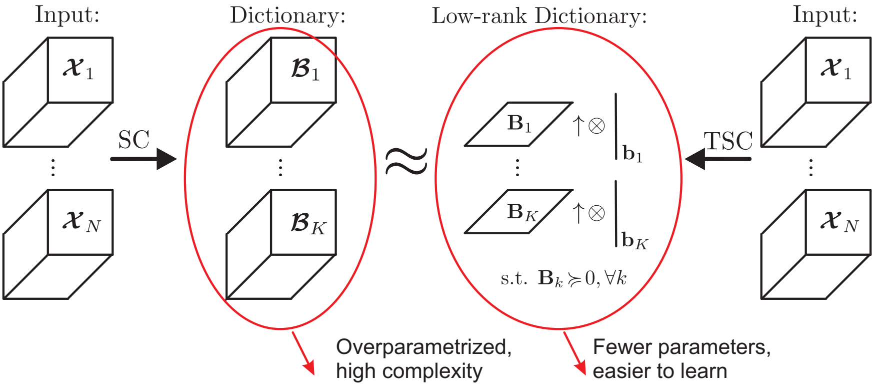
- However, $\boldsymbol{\mathcal{B}}$ have three modes (overparametrised model), so we learn instead low-rank dictionary:

$$\underset{\substack{\mathbf{B}_1,\ldots,\mathbf{B}_K\\ \mathbf{b}_1,\ldots,\mathbf{b}_K\\ \boldsymbol{\alpha}^1,\ldots,\boldsymbol{\alpha}^N}}{\arg\min} \sum_{n=1}^{N} \left\| \boldsymbol{\mathcal{X}}_n - \sum_{k=1}^{K} \left(\mathbf{B}_k \uparrow\!\otimes \mathbf{b}_k\right) \alpha_k^n \right\|_F^2 + \lambda \left\|\boldsymbol{\alpha}^n\right\|_1. \tag{13}$$

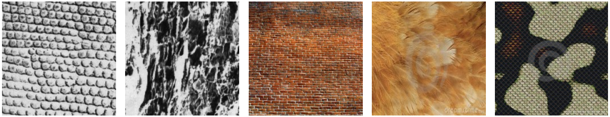
- Resulting sparse codes $\boldsymbol{\alpha}$ are pooled and used for SVM training.

Sparse Coding for Third-order Tensor Descriptors (TSC)

- We use training set of TOSST descriptors $\boldsymbol{\mathcal{X}}_1, ..., \boldsymbol{\mathcal{X}}_N$.
- We learn low-rank dictionary atoms $\mathbf{B}_1 \uparrow\otimes \mathbf{b}_1, ..., \mathbf{B}_K \uparrow\otimes \mathbf{b}_K$ (outer product of matrices with vectors).
- They approximate full-rank tensor atoms $\boldsymbol{\mathcal{B}}_1, ..., \boldsymbol{\mathcal{B}}_K$.



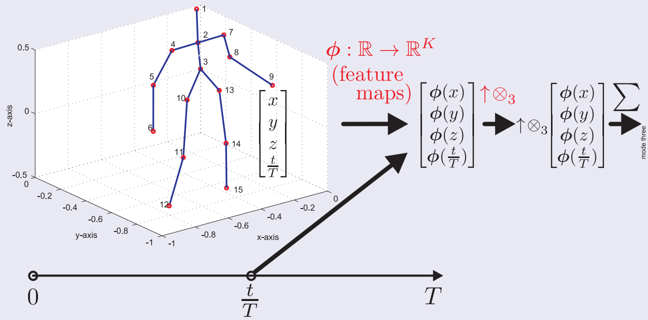
Results

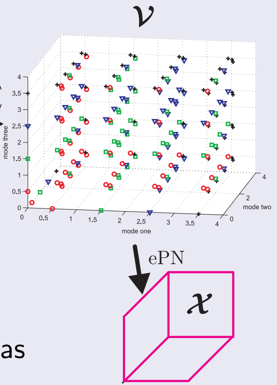


- Brodatz textures; 99.9% accuracy (the state of the art); others score $\sim$ 98.72%.
- UIUC materials recognition; 58.0% accuracy.
- PASCAL VOC07 descriptor compression: 61.2% mAP (25K signature) vs. 61.3% mAP (176K signature).

Action Recognition from 3D Skeletons

Sequence Compatibility Kernel

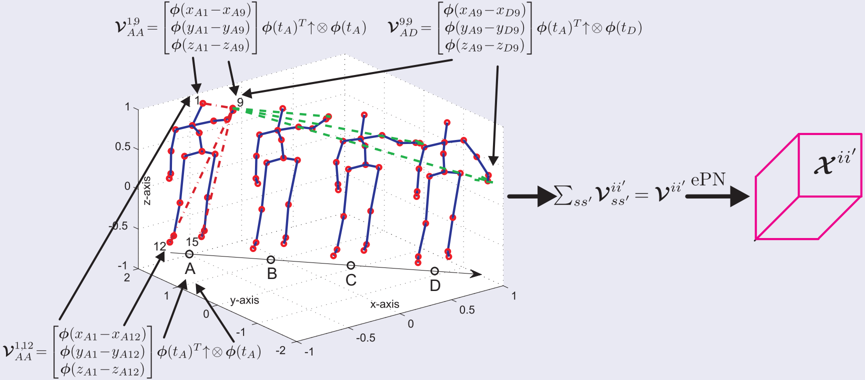




- Components $\phi(x)$, $\phi(y)$, $\phi(z)$, $\phi(\frac{t}{T})$ denoted as $(\circ, \square, \triangledown, +)$.
- $\mathcal{V}$ captures all triplets: $(\circ\square\triangledown)$, $(\circ\square+)$, $(\circ\triangledown+)$, $(\square\triangledown+)$.
- ePN evens out counts of these co-occurrences.
- Tensors $\mathcal{X}$ are the samples for training SVM.

Action Recognition from 3D Skeletons

Dynamics Compatibility Kernel



- Enumerate all unique joint displacement vectors $\mathbf{x}_{it}-\mathbf{x}_{jt',i\leq j,t\leq t'}$.
- Embed displacements into RKHS and linearise to obtain $\phi(\mathbf{x}_{it}-\mathbf{x}_{jt'})$.
- Embed start-/end-times into RKHS, linearise to obtain $\phi(\frac{t}{T})$, $\phi(\frac{t'}{T})$.
- Take outer products $\phi(\mathbf{x}_{it}-\mathbf{x}_{jt'})\phi(\frac{t}{T})\uparrow\otimes\phi(\frac{t'}{T})$, aggregate+ePN.

Eigenvalue Power Normalisation

- Four simple steps in MATLAB:

$$(\boldsymbol{\mathcal{E}}; \boldsymbol{A}_1, ..., \boldsymbol{A}_r) = \text{HOSVD}(\boldsymbol{\mathcal{V}}) \tag{14}$$

$$\hat{\boldsymbol{\mathcal{E}}} = \text{Sgn}(\boldsymbol{\mathcal{E}})\, |\boldsymbol{\mathcal{E}}|^{\gamma} \tag{15}$$

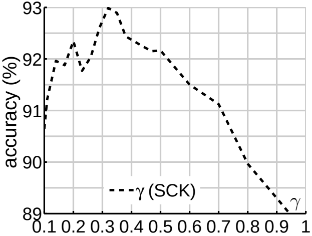
$$\hat{\boldsymbol{\mathcal{V}}} = ((\hat{\boldsymbol{\mathcal{E}}} \otimes_1 \boldsymbol{A}_1) \ldots) \otimes_r \boldsymbol{A}_r \tag{16}$$

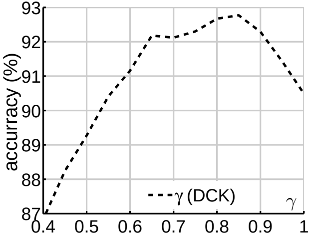
$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}}(\boldsymbol{\mathcal{V}}) = \text{Sgn}(\hat{\boldsymbol{\mathcal{V}}})\, |\hat{\boldsymbol{\mathcal{V}}}|^{\gamma^*} \tag{17}$$

- Perform Higher Order SVD (equivalent of SVD for more than 2 modes).
- Obtain the core tensor $\boldsymbol{\mathcal{E}}$ (equivalent of singular values).
- Power-normalise this spectrum (values $\boldsymbol{\mathcal{E}}$ can be negative).
- Assemble back tensor, if needed, perform additionally standard PN.

Action Recognition from 3D Skeletons: results

- Eigenvalue Power Normalisation w.r.t. γ:





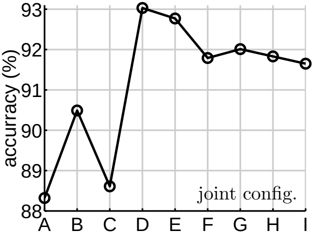
Action Recognition from 3D Skeletons: results

- Florence3D-Action (state-of-the-art):

	SCK	DCK		SCK+DCK
accuracy	92.98%	93.03%	92.77%	**95.47**%
size	26565	9450	16920	43485

Bag-of-Poses 82.00%	*SE*(3) 90.88%	

- Using fewer less noisy key-joints may be better (SCK):



Action Recognition from 3D Skeletons: results

- UTKinect-Action (state-of-the-art):

	SCK	DCK	SCK+DCK
accuracy	96.08%	97.69%	**98.39**%
size	40480	16920	57400

3D joints hist. 90.92%	$SE(3)$ 97.08%	

- MSR-Action3D:

	SCK+DCK	$SE(3)$
accuracy, standard protocol	**92.7**%	89.48%
accuracy, specific classes/subjectss	**96**%	92.46%
size	57400	-

Natural Inner Product on Gaussians

- Gaussian kernel between $\mathbf{u} \in \mathbb{R}^{d'}$ and $\bar{\mathbf{u}} \in \mathbb{R}^{d'}$ can be simply rewritten as:

$$G_\sigma(\mathbf{u}-\bar{\mathbf{u}}) = e^{-\|\mathbf{u}-\bar{\mathbf{u}}\|_2^2/2\sigma^2} = \left(\frac{2}{\pi\sigma^2}\right)^{\frac{d'}{2}} \int_{\zeta\in\mathbb{R}^{d'}} G_{\sigma/\sqrt{2}}(\mathbf{u}-\zeta)\, G_{\sigma/\sqrt{2}}(\bar{\mathbf{u}}-\zeta)\,\mathrm{d}\zeta. \tag{18}$$

- Finite approximation by $\zeta_1, ..., \zeta_Z$ pivots is given by:

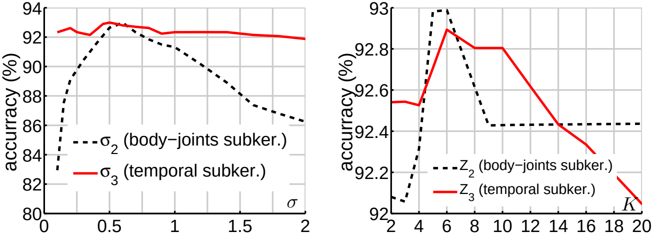
$$\phi(\mathbf{u}) = \left[G_{\sigma/\sqrt{2}}(\mathbf{u}-\zeta_1), ..., G_{\sigma/\sqrt{2}}(\mathbf{u}-\zeta_Z)\right]^T, \tag{19}$$

$$\text{and } G_\sigma(\mathbf{u}-\bar{\mathbf{u}}) \approx \left\langle \frac{\phi(\mathbf{u})}{\|\phi(\mathbf{u})\|_2}, \frac{\phi(\bar{\mathbf{u}})}{\|\phi(\bar{\mathbf{u}})\|_2} \right\rangle. \tag{20}$$

- As few as 6 pivots yield $\leq 0.8\%$ approximation error.

Action Recognition from 3D Skeletons: results

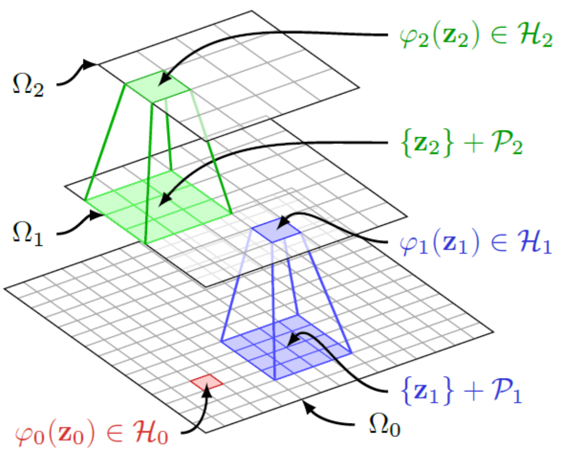
- Florence3D-Action w.r.t. kernel radii and pivot numbers:



- Not too sensitive to parameter variations.
- More pivots provide better RBF approximation but SVM has to learn more parameters (overfitting).

Extensions to Deep Architectures

- Convolutional Kernel Networks: future talk.



- The simplest extension - instead of Fisher Vectors apply HOSVD+ePN to aggregate over multiple CNN-based sub-patches.

Conclusions

- It may look difficult, but it requires few easy steps only in practice:
  A. Extract your favourite features.
  B. Embed them into RKHS/linearise.
  C. Form outer products of desired order.
  D. Aggregate and apply ePN.
  E. Train SVM (or any favourite classifier).
- Interested in any related ideas? Talk to me to see if we can collaborate :-) Ideas take know-how, time and people to develop them.
- References:

P. Koniusz, F. Yan, P. H. Gosselin, K. Mikolajczyk.
Higher-order Occurrence Pooling for Bags-of-Words: Visual Concept Detection, TPAMI, 2016.

P. Koniusz, A. Cherian.
Sparse Coding for Third-order Super-symmetric Tensor Descriptors with Application to Texture Recognition, CVPR, 2016.

P. Koniusz, A. Cherian, F. Porikli.
Tensor Representations via Kernel Linearlization for Action Recognition from 3D Skeletons, ECCV, 2016.

J. Mairal, P. Koniusz, Z. Harchaoui, C. Schmid.
Convolutional Kernel Networks, NIPS, 2014.

Thank You