Channel Representations for Machine Learning

Michael Felsberg and Klas Nordberg

Department of Electrical Engineering
Computer Vision Laboratory
Linköping University

Artificial Visual Systems (AVS)

- Huge progress in computer vision / AVS, but myriad of open problems
- Robustness in unconstrained environments, e.g. visual analysis of plants
- Prediction of actions in collaboration with humans, e.g. ADAS
- Principle: if something does not work (sufficiently well), look at some working system!
- Conclusion: for many applications design AVS similar to HVS
- BUT without forgetting about established methods/concepts from science and engineering

[Pauli, A., Olivar, A., Frontiers in Computer Vision, white paper 2010]

Representations and Learning

- Unlike deep learning: design descriptor as a basis for learning
- Experience: probabilistic models central to success in computer vision
- Choice here: non-parametric models
  - Kernel Density Estimator (KDE) / Parzen window approach:
    continuous outputs, BUT: lots of data result in slow read out
  - Histograms / vector quantization / bag of words:
    very efficient, BUT: binned data / quantization effects

Combining Histograms and KDEs

Goal: combine advantages of

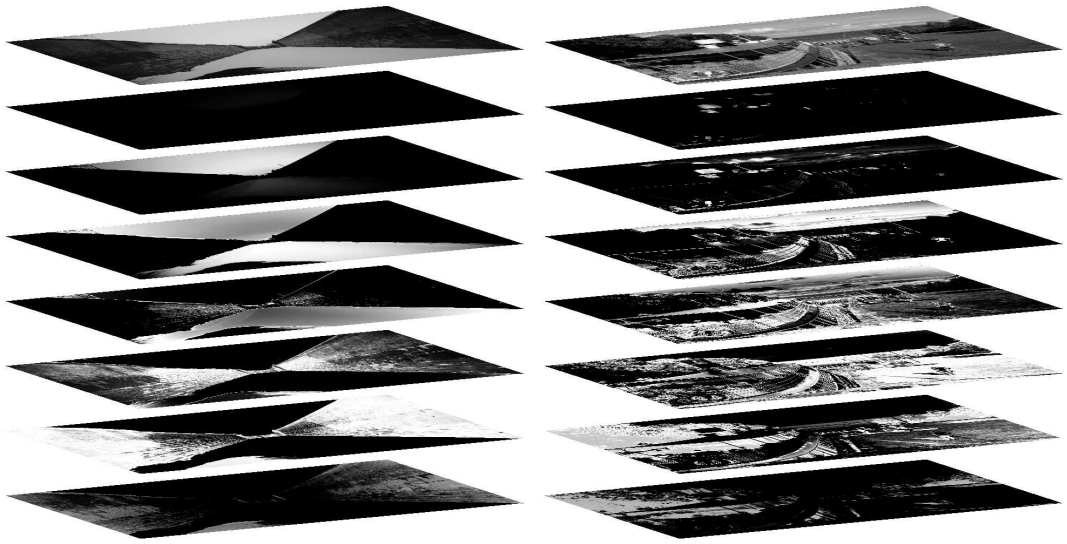
- histograms (fast readout)
- KDEs (no quantization)

Approach:

- sampling of KDE (discrete density)
- soft histogram bins
- from soft histogram: maximum likelihood estimate (MLE)

Distribution Field (DF)

explode image [Sevilla-Lara&Learned-Miller, CVPR'12]



Distribution Field (DF)

Smoothed local histogram

1. Image values $I(i,j)$ quantized into $b = 16$ levels k

$$d(i,j,k) = \begin{cases} 1 & \text{if } I(i,j) == k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

2. Spatial smoothing (pooling) with 2D Gaussian kernels $h_{\sigma_s}(i,j)$ at two scales $\sigma_s = 1$ and $\sigma_s = 2$

$$d_s(i,j,k) = (d(\cdot,\cdot,k) * h_{\sigma_s})(i,j) \quad \text{for all } i,j \quad (2)$$

3. Histogram smoothing with 1D Gaussian kernel $h_{\sigma_f}(k)$ with $\sigma_f = 0.625$

$$d_{ss}(i,j,k) = (d_s(i,j,\cdot) * h_{\sigma_f})(k) \quad \text{for all } k \quad (3)$$

Channel Representation [Granlund, AFPAC 2000]

Distribution Field

1. one-out-of b coding
2. spatial pooling (histogram)
3. smoothing of histogram

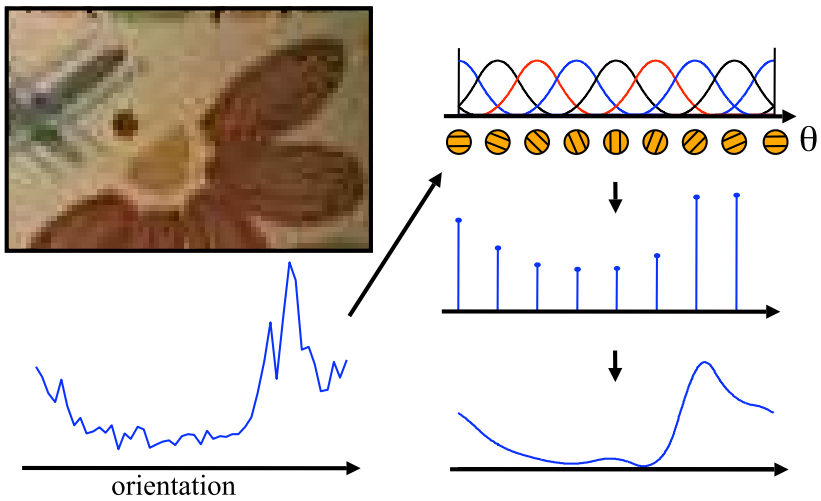
Result: smoothed histogram
Modes biased

Channel Representation

1. one-out-of b coding
2. soft assignment (three bins)
3. spatial pooling (histogram)

Result: histogram of soft assignments
MLE [Felsberg et al., Frontiers in Robotics and AI 2015]

Channel Representation [Granlund, AFPAC 2000]

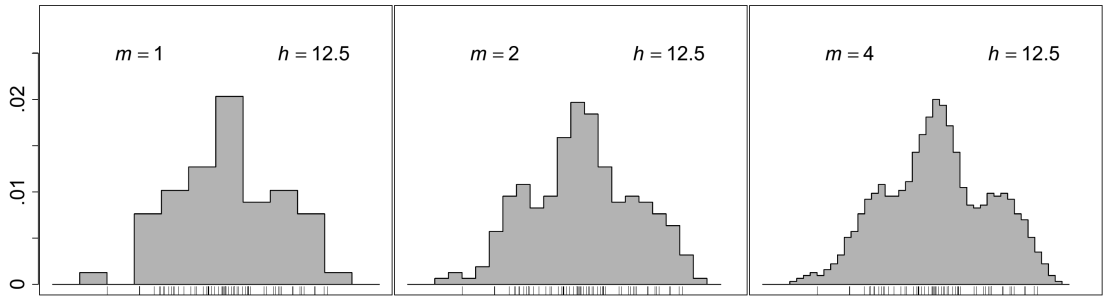


- Population codes [Pouget et al., Nature Reviews 2000]
- Channel-codes [Snippe&Koenderink, Biol. Cyb. 1992]
- Averaged shifted histogram, ASH [Scott, AStat. 1985]

Averaged Shifted Histogram

ASH means to average m shifted histograms $\hat{f}_1(x), \ldots, \hat{f}_m(x)$ with bin-width h

$$\hat{f}_{\mathrm{ASH}}(x; m) = \frac{1}{m} \sum_{i=1}^{m} \hat{f}_i(x) \tag{4}$$



Averaged Shifted Histogram

or equivalently based on fine histogram, piece-wise constant in intervals of width h/m: $\hat{g}(x)$

$$\hat{f}_{\mathrm{ASH}}(x; m) = \frac{1}{m} \sum_{i=1-m}^{m-1} \left(1 - \frac{|i|}{m}\right) \hat{g}(x+i) \tag{5}$$

ASH is obtained by convolving $\hat{g}$ with a triangle-kernel of width $2m-1$,
or in general with kernel $w_m(i) \geq 0$

$$\hat{f}_{\mathrm{ASH}}(x; m) = \sum_{i=1-m}^{m-1} w_m(i) \hat{g}(x+i) \tag{6}$$

Averaged Shifted Histogram

In the limit, the ASH becomes a kernel density estimator [Scott, Wiley 1992]

$$\lim_{m\to\infty} \hat{f}_{\mathrm{ASH}}(x;m) = \frac{1}{nh}\sum_{j=1}^{n} K\left(\frac{x-x_j}{h}\right) \qquad (7)$$

In (6), we can replace x with the respectively closest bin-center (sampling) without changing the value. Since w/K is band-limited, we may keep the sample frequency also for $m \to \infty$.

Definitions

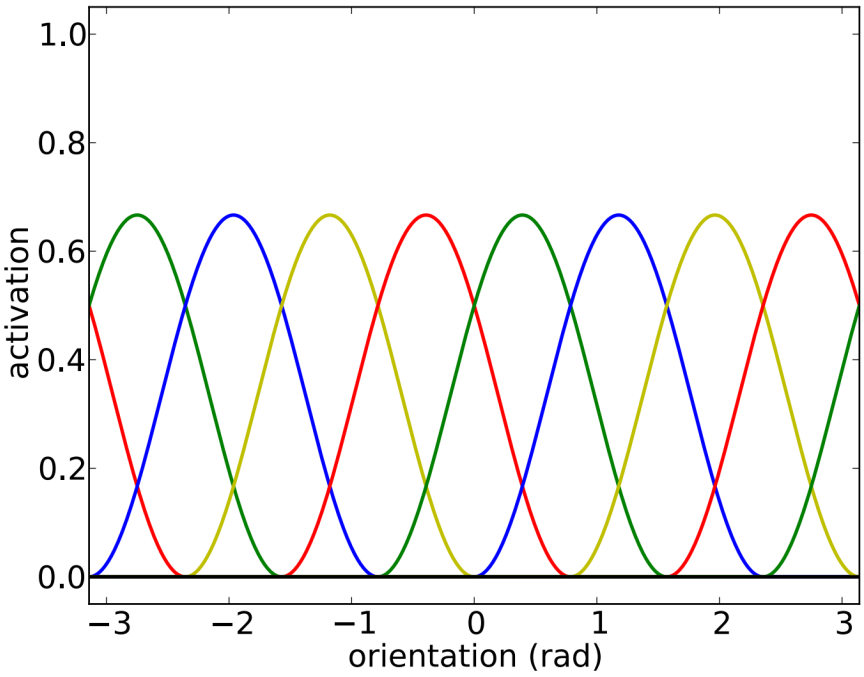
- n measurements x_j
- Basis function / kernel K
- Width parameter h
- Channel coefficient

$$c_k = \frac{1}{nh}\sum_{j=1}^{n} K(x_j/h - k) \qquad k \in \mathbb{N} \qquad (8)$$

- Channel vector

$$\mathbf{c} = \{c_k(x)\}_{k\in\mathbb{N}} \qquad (9)$$

Illustration



Kernel Functions

- Population codes: $\cos^+$ [Pouget et al., Annu. Rev. Neurosci. 2003]
- Channel representation: $\cos^2$ [Granlund, AFPAC 2000]
- Channel representation: quadratic B-splines [Felsberg et al., TPAMI 2006]
- Channel representation: truncated Gaussian function [Forssén, PHD 2004]
- P-channels: rectangle + linear B-spline [Felsberg&Granlund, ICPR 2006]
- Channel-Coded Feature Maps: mono-pieces [Jonsson&Felsberg, IMAVIS 2009]

Two choices

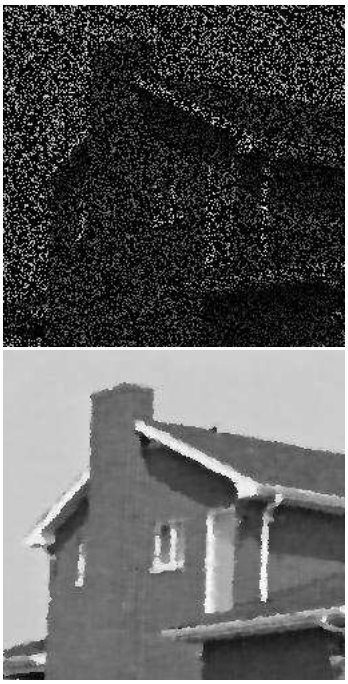
- $\cos^2$ Channels (overlap 3, $h = 3$)

$$K(x) = \frac{2}{3}\cos^2\frac{x\pi}{3} \qquad x \in [-\frac{3}{2};\frac{3}{2}] \qquad (10)$$

- Quadratic B-splines (overlap 3, $h = 3$)

$$K(x) = \begin{cases} 3/4 - x^2 & |x| \leq 1/2 \\ (|x| - 3/2)^2/2 & 1/2 < |x| \leq 3/2 \\ 0 & \text{otherwise} \end{cases} \qquad (11)$$

Channel Smoothing [Felsberg et al., TPAMI 2006]



Properties of kernel functions

- Kernels must integrate to 1: $\int_x K(x)\,dx = 1$
- Channel coefficients must sum to 1:

$$\sum_k c_k(x) = 1 \qquad \text{for all } x \tag{12}$$

- For $\cos^2$: sum of squared channel coefficients is constant:

$$\sum_k c_k^2(x) = \frac{1}{2} \qquad \text{for all } x \tag{13}$$

- Is the $\cos^2$ kernel unique?

Uniqueness Proof

Theorem

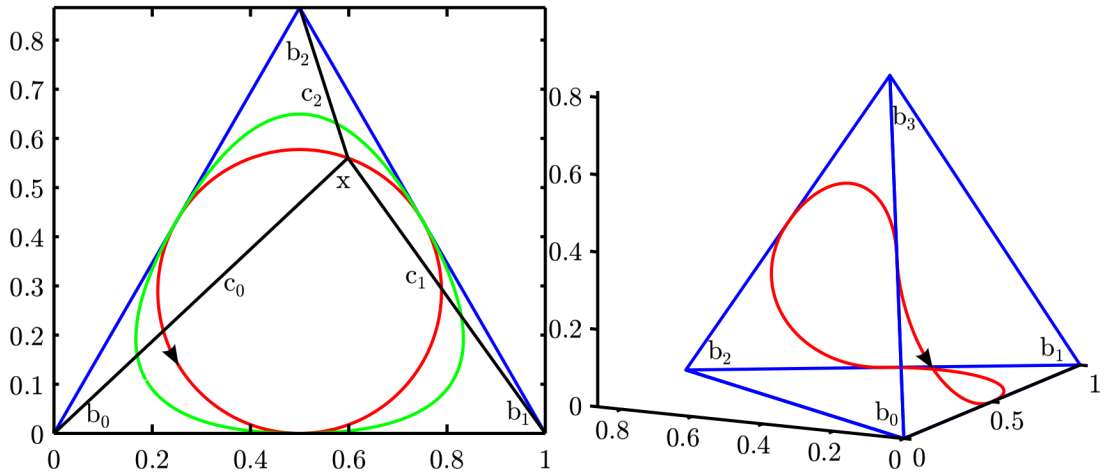
(Minimum overlap channel basis) The minimum overlap of a channel basis with constant l_2-norm $\|\mathbf{c}\|_2 = \mu$ for all x is 3. $\mu = \frac{1}{2}$.

Theorem

(Uniqueness of $\cos^2$ kernel) The unique channel basis function with minimal overlap and integer spacing is given by (10).

[Felsberg et al., Frontiers in Robotics and AI, 2015]

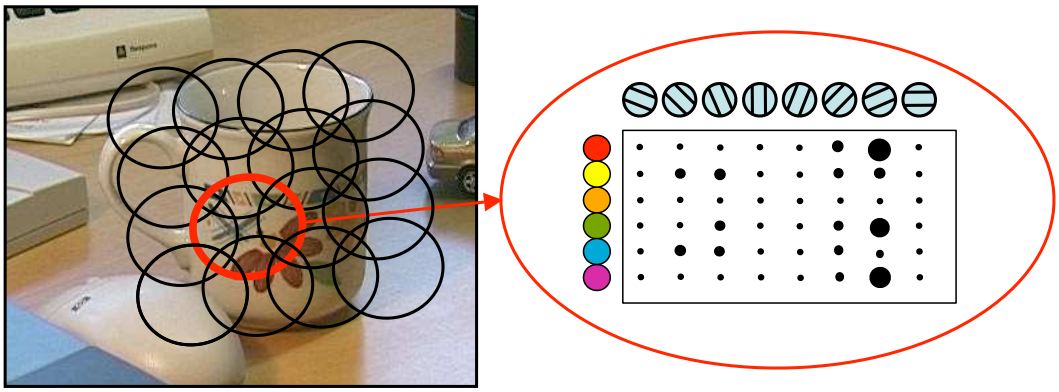
Geometric Interpretation



Curves with constant distance to origin (constant l_2-norm) on the surface of a simplex (constant l_1-norm)

Channel Coded Feature Maps (CCFM)

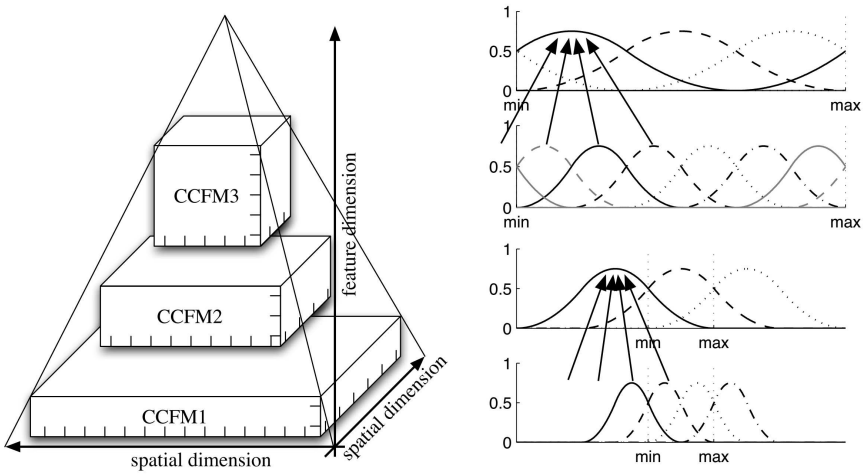
Idea: channel encode image value(s) / feature(s) *and* position coordinates



[Felsberg&Granlund, ICPR 2006]
[Jonsson&Felsberg, IMAVIS 2009]

Channel Coded Feature Maps (CCFM)

- Simultaneous pooling of features (SIFT, HOG)
- and position parameters
- Uncertainty $(\Delta x)(\Delta f) \geq k$ [Felsberg, SSVM2009]



Distribution Field Tracking (DFT)

[Sevilla-Lara&Learned-Miller, CVPR 2012]
During tracking: local search for optimal l_1 fit between DF of template d_{model} and DF of local window d_f

$$l_1(d_{\text{model}}, d_f) = \sum_{i,j,k} |d_{\text{model}}(i,j,k) - d_f(i,j,k)| \qquad (14)$$

At the local optimum, the template $d_{\text{model},t}$ is updated using the current DF d_f

$$d_{\text{model},t+1}(i,j,k) = \lambda d_{\text{model},t}(i,j,k) + (1-\lambda) d_f(i,j,k) \qquad (15)$$

where $\lambda = 0.95$

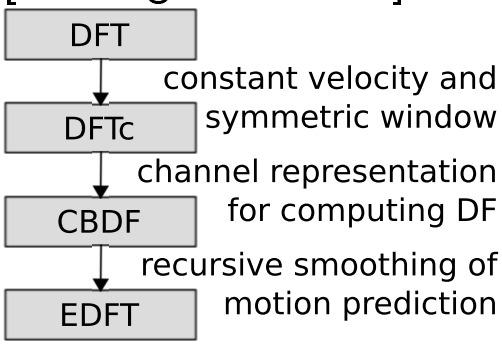
Enhanced Distribution Field Tracking

- Symmetric search window
- DFs replaced with Channel Representations

$$h = \frac{4\pi}{3}\sqrt{\frac{91}{\pi^2 - 6}} \approx 20.31$$

- Smoothed motion prediction

[Felsberg, VOT 2013]



$$\mathbf{m}_{p,\text{new}} = \frac{1}{2}(\mathbf{m}_{p,\text{old}} + \mathbf{p}_{\text{new}} - \mathbf{p}_{\text{old}}) \qquad (16)$$

Extending EDFT: Non-Linear Update

$$\mathbf{C}_{\text{model},t} = \left((1-\gamma)\mathbf{C}^q_{\text{model},t-1} + \gamma\mathbf{C}^q_f\right)^{\frac{1}{q}} \qquad (17)$$

- Increasing q shifts the weight towards larger elements
- If the current coefficient is larger than the model coefficient: faster adaptation
- If the model coefficient is larger than the current coefficient: slower forgetting
- Increasing γ: faster learning and forgetting
- For $q \to \infty$, we obtain the maximum rule $\mathbf{C}_{\text{model},t} = \max(\mathbf{C}_{\text{model},t-1}, \mathbf{C}_f)$

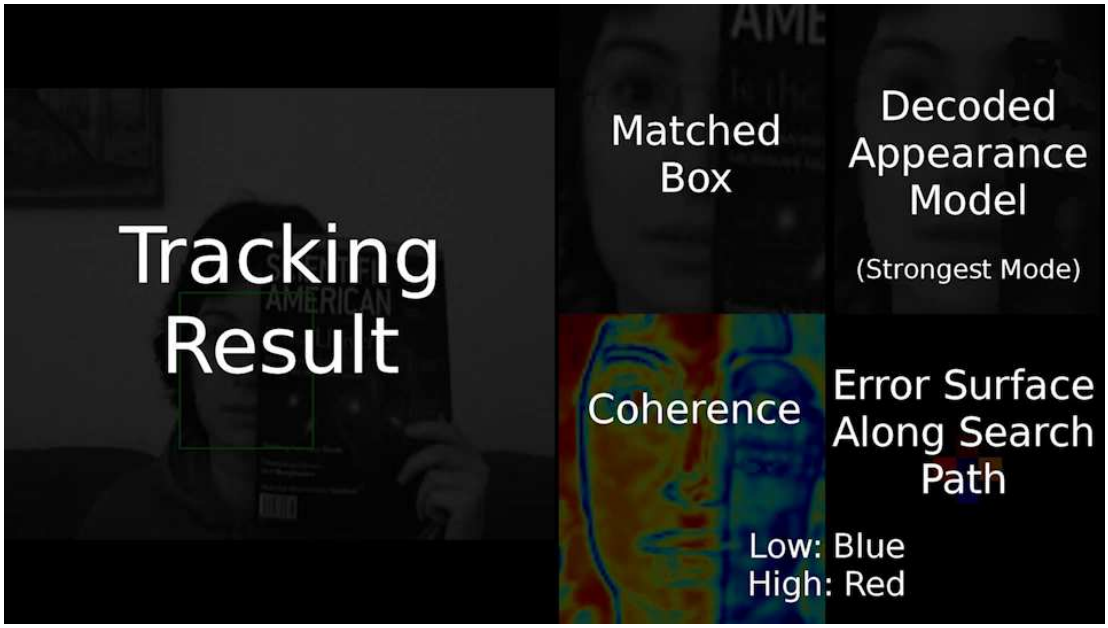
[Öfjäll&Felsberg, VOT 2014]

VOT 2014 Region Noise Results by EDFT Variants

Method	Mean		Median	
	Accuracy	Robustness	Accuracy	Robustness
NCC	0.456	3.0	0.414	1.8
DFT	0.493	2.4	0.512	2.4
EDFT	0.486	2.0	0.486	1.9
qEDFT (q=5)	0.502	2.0	0.512	**1.3**
cEDFT	0.489	2.1	0.492	1.9
qcEDFT (q=4)	0.509	**1.7**	0.502	1.5
maxcEDFT	0.514	2.2	0.500	2.0
qσEDFT (q=4)	**0.521**	1.9	**0.534**	1.6
maxσEDFT	0.516	2.1	0.529	2.0

[Öfjäll&Felsberg, VOT 2014]

Example Videos



[Felsberg, VOT 2013; Öfjäll&Felsberg, VOT 2014]

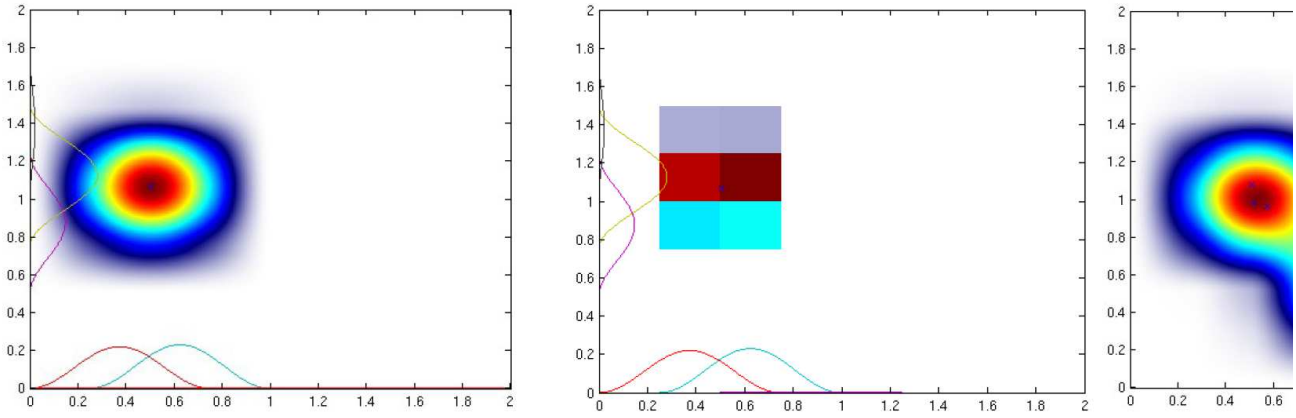
Optical Flow

- Channel Representations have also been used for optical flow analysis
- Replace brightness constancy with channel constancy [Sevilla-Lara et al., ECCV 2014]
- Results on Sintel dataset [Butler et al., ECCV 2012]
-

	tched
	3.781
	2.064
	2.344
	4.509

Associative Mappings

[Granlund, Invited Talk AFPAC 2000]



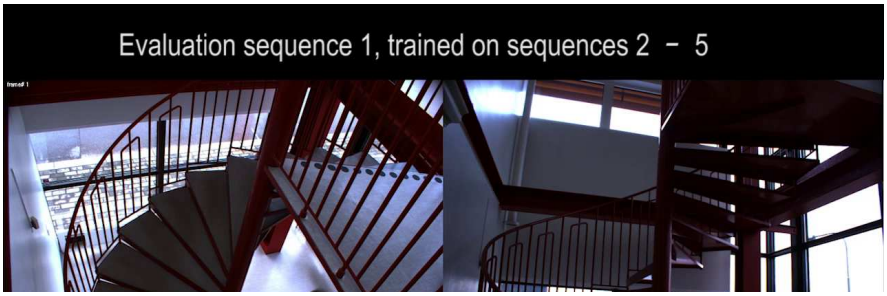
Learning Associative Mappings

- Batch method [Granlund, Invited Talk AFPAC 2000]

$$\mathbf{A}^* = \arg\min_{\mathbf{A}} \|\mathbf{C}_\mathrm{O} - \mathbf{A}\mathbf{C}_\mathrm{I}\|_F^2 \qquad \text{s.t. } a_{ij} \geq 0$$

- Online method using Neyman's chi-square divergence D_{-1} [Felsberg et al., TPAMI 2013]

$$\mathbf{A}^* = \arg\min_{\mathbf{A}} D_{-1}[\mathbf{C}_\mathrm{O} \| \mathbf{A}\mathbf{C}_\mathrm{I}]$$



qHebbian Learning

- Hebbian learning suffers from unbounded coefficients
- Avoided by normalization (forgetting factor)
- Non-linear functions (as used in the adaptive model) may be applied if Hebbian learning is applied to channel vectors

$$\mathbf{A}_t = \left((1-\gamma)\mathbf{A}_{t-1}^q + \gamma\mathbf{B}_t^q\right)^{\frac{1}{q}}$$

- where $\mathbf{B} = \mathbf{c}_\mathrm{O}\mathbf{c}_\mathrm{I}^T$
- $q \to \infty$ results in maxHebb

[Öfjäll&Felsberg, BMVC 2014]

Results

[Öfjäll&Felsberg, BMVC 2014]

Questions?