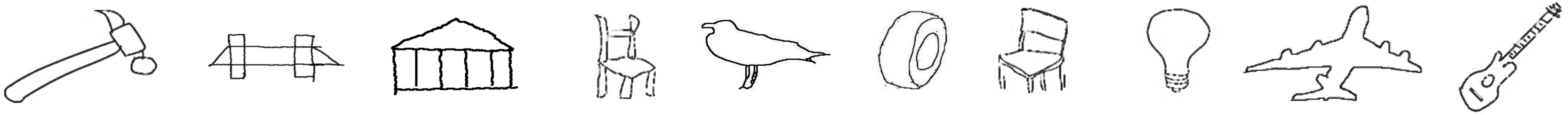


Asymmetric feature maps with application to sketch-based image retrieval

Giorgos Tolias
Center for Machine Perception

joint work with Ondřej Chum

Overview

- Explicit feature maps
- Efficient match kernels
- Trigonometric polynomial scores
- Asymmetric feature maps
- Sketch-based image retrieval

Explicit feature maps

- Stationary kernels
- Fourier series approximation
- Low dimensional feature maps

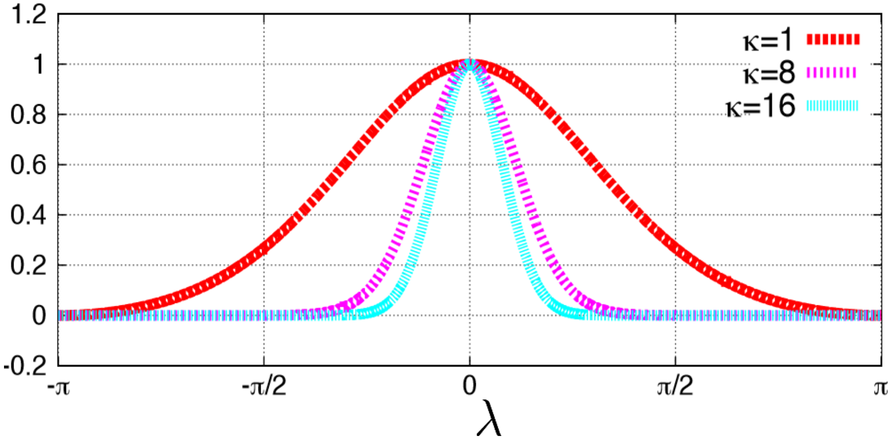
Explicit feature maps

- Stationary kernel that depends on the difference $\lambda = \theta_1 - \theta_2$

$$k(\theta_1, \theta_2) = k(\theta_1, \theta_1 - \lambda) = k(\lambda)$$

- Construct a mapping: $\theta \to \Psi(\theta), \; \Psi(\theta) \in \mathbb{R}^n$
  - s.t. inner product approximates a non-linear kernel

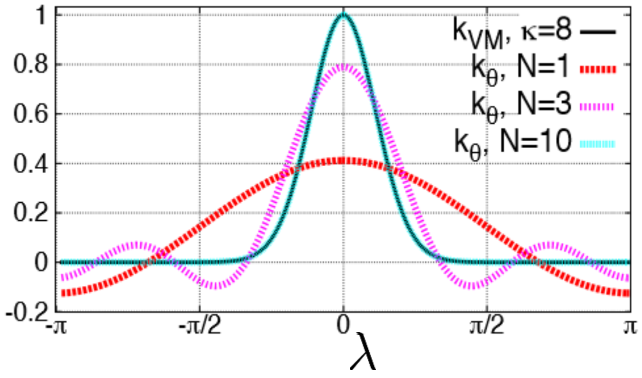
$$\Psi(\theta_1)^\top \Psi(\theta_2) \approx k(\lambda)$$



Fourier series approximation

- Fourier series approximation with *N* frequencies

$$k(\lambda) \approx \sum_{n=\{0,1,\ldots,N\}} \alpha_n \cos(n\lambda) = \sum_{n=\{0,1,\ldots,N\}} \alpha_n \cos(n(\theta_1 - \theta_2))$$



- Employ trigonometric identity

$$\cos(\theta_1 - \theta_2) = \cos(\theta_1)\cos(\theta_2) + \sin(\theta_1)\sin(\theta_2) = (\cos(\theta_1), \sin(\theta_1))(\cos(\theta_2), \sin(\theta_2))^\top$$

- Construct embedding with 2N+1 dimensions

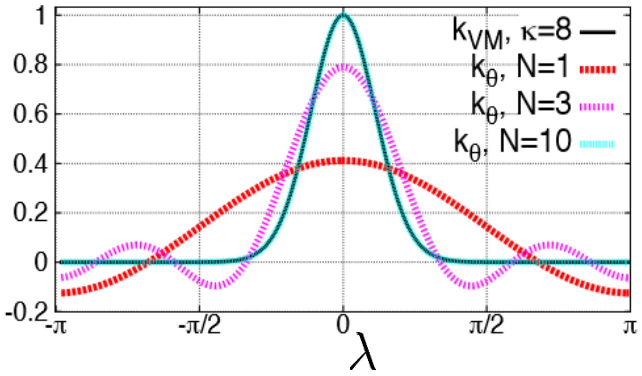
$$\Psi(\theta) = (\sqrt{\alpha_0}, \sqrt{\alpha_1}\cos(\theta), \sqrt{\alpha_1}\sin(\theta), \ldots, \sqrt{\alpha_N}\cos(N\theta), \sqrt{\alpha_N}\sin(N\theta))^\top$$

[Vedaldi & Zisserman '12]

Fourier series approximation

- Fourier series approximation with *N* frequencies

$$k(\lambda) \approx \sum_{n=\{0,1,\ldots,N\}} \alpha_n \cos(n\lambda) = \sum_{n=\{0,1,\ldots,N\}} \alpha_n \cos(n(\theta_1 - \theta_2))$$



- Employ trigonometric identity

$$\cos(\theta_1 - \theta_2) = \cos(\theta_1)\cos(\theta_2) + \sin(\theta_1)\sin(\theta_2) = \overset{\Psi(\theta_1)}{(\cos(\theta_1), \sin(\theta_1))}(\cos(\theta_2), \sin(\theta_2))^\top$$

- Construct embedding with 2N+1 dimensions

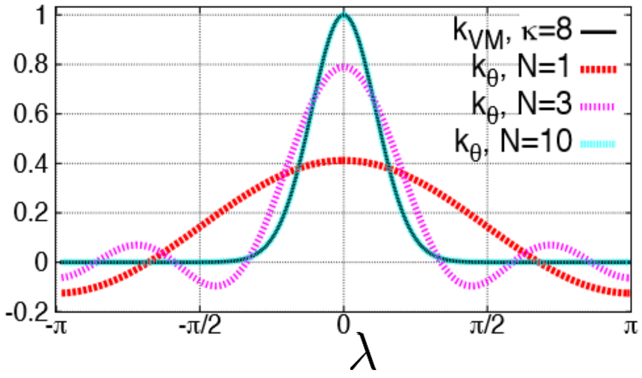
$$\Psi(\theta) = (\sqrt{\alpha_0}, \sqrt{\alpha_1}\cos(\theta), \sqrt{\alpha_1}\sin(\theta), \ldots, \sqrt{\alpha_N}\cos(N\theta), \sqrt{\alpha_N}\sin(N\theta))^\top$$

[Vedaldi & Zisserman '12]

Fourier series approximation

- Fourier series approximation with N frequencies

$$k(\lambda) \approx \sum_{n=\{0,1,\ldots,N\}} \alpha_n \cos(n\lambda) = \sum_{n=\{0,1,\ldots,N\}} \alpha_n \cos(n(\theta_1 - \theta_2))$$



- Employ trigonometric identity

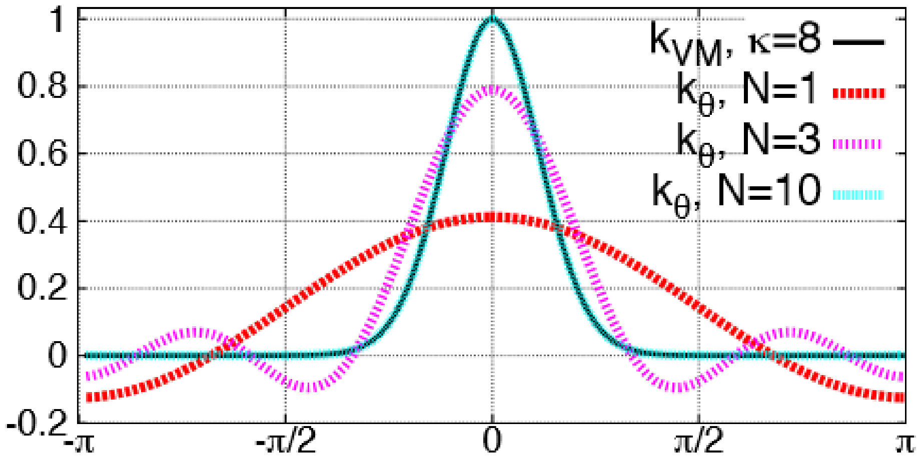
$$\cos(\theta_1 - \theta_2) = \cos(\theta_1)\cos(\theta_2) + \sin(\theta_1)\sin(\theta_2) = \overbrace{(\cos(\theta_1), \sin(\theta_1))}^{\Psi(\theta_1)}\underbrace{(\cos(\theta_2), \sin(\theta_2))^\top}_{\Psi(\theta_2)}$$

- Construct embedding with 2N+1 dimensions

$$\Psi(\theta) = (\sqrt{\alpha_0}, \sqrt{\alpha_1}\cos(\theta), \sqrt{\alpha_1}\sin(\theta), \ldots, \sqrt{\alpha_N}\cos(N\theta), \sqrt{\alpha_N}\sin(N\theta))^\top$$

[Vedaldi & Zisserman '12]

Fourier series approximation

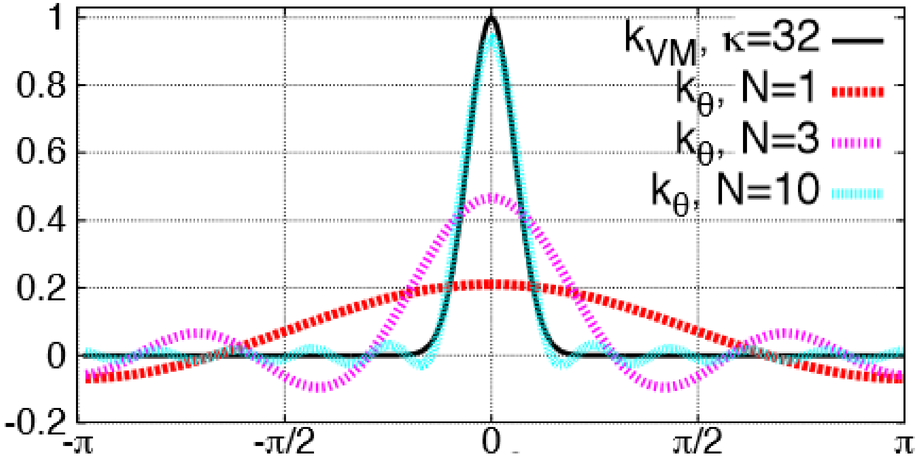


More frequencies

→ better approximation

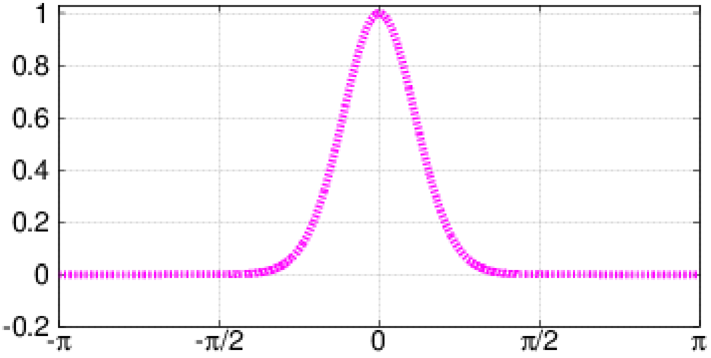
Higher selectivity

→ more frequencies needed



Numeric example

- $\theta_1 = 0.5$
- $\theta_2 = 1.2$
- $k_{VM}(\theta_{1,}\ \theta_2) = 0.1691$



- map θ_1 to 11D vector:

$$x_1 = [0.3989\quad 0.4788\quad 0.2668\quad 0.0297\quad -0.1393\quad -0.2022\quad 0.2616\quad 0.4156\quad 0.4182\quad 0.3044\quad 0.1510]^T$$

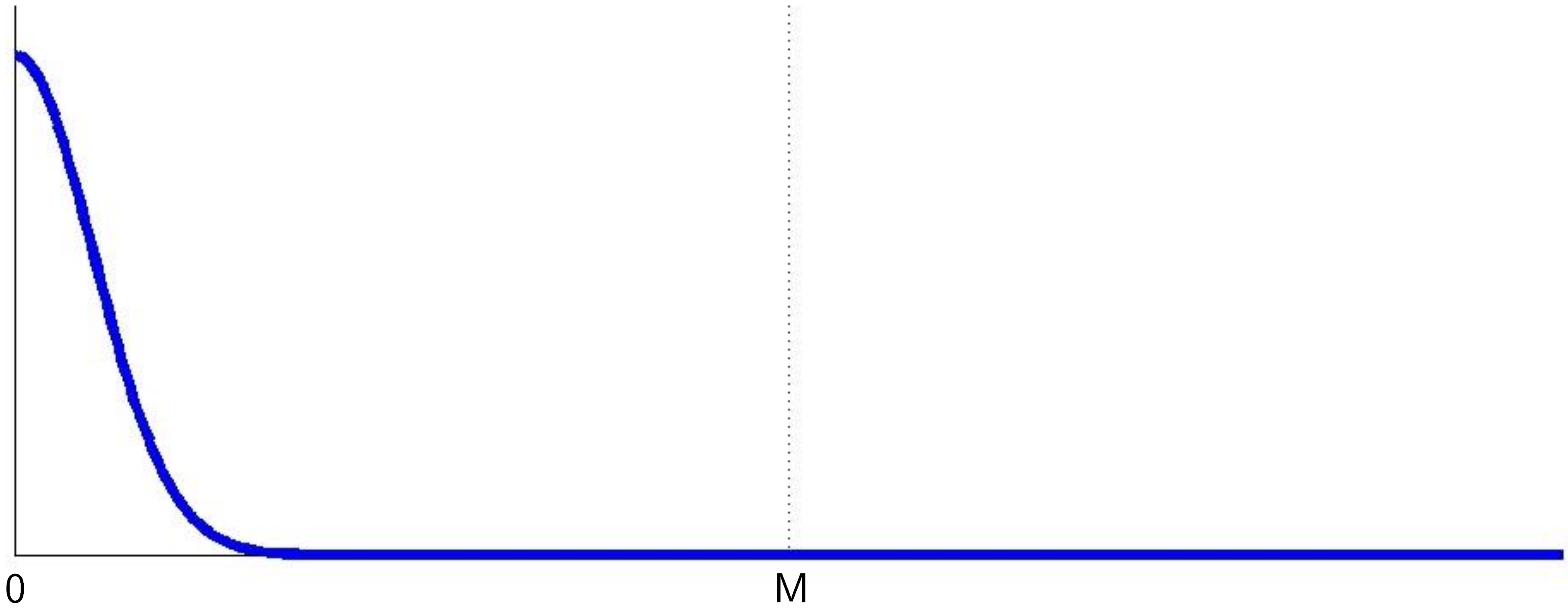
- map θ_2 to 11D vector:

$$x_2 = [0.3989\quad 0.1977\quad -0.3642\quad -0.3759\quad 0.0293\quad 0.2423\quad 0.5085\quad 0.3336\quad -0.1855\quad -0.3335\quad -0.0705]^T$$

- $x_1^T\ x_2 = 0.1743$

Example: Approximating 1D RBF

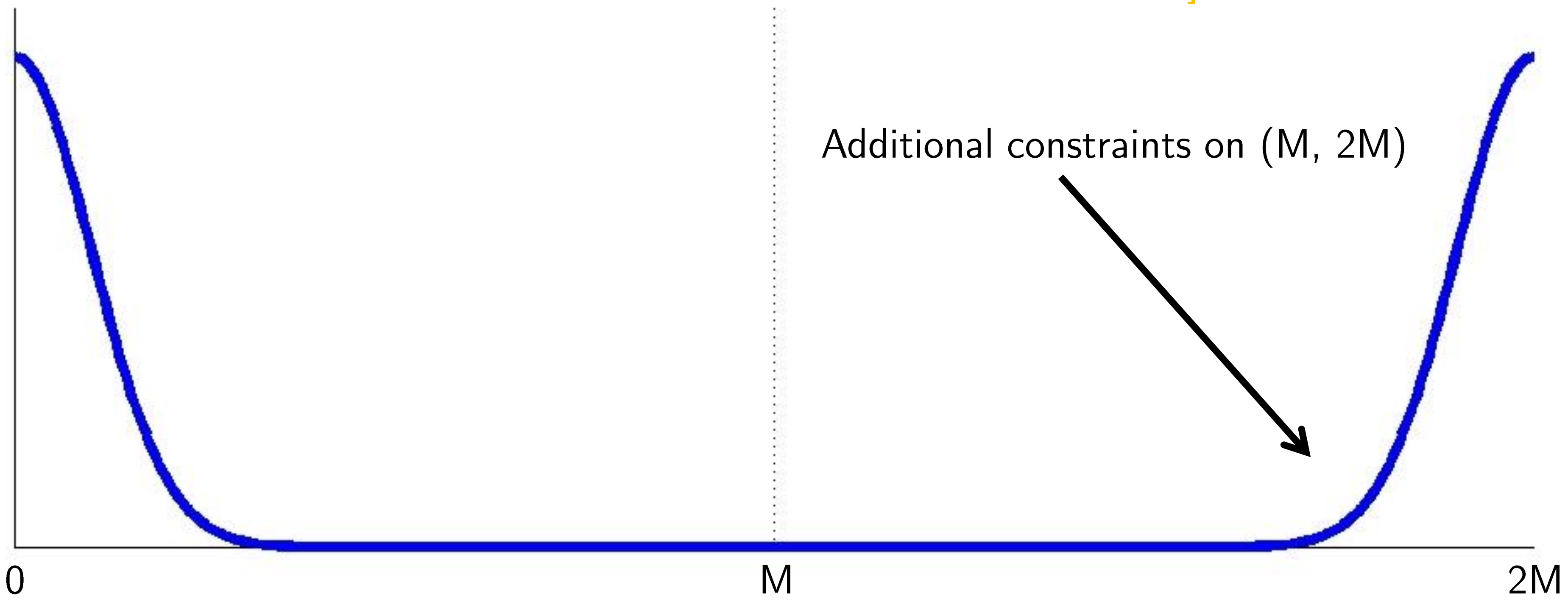
Approximate Gaussian on interval [0, M]



Approximation using harmonic frequencies

Approximate a periodic function on interval [0, 2M] instead

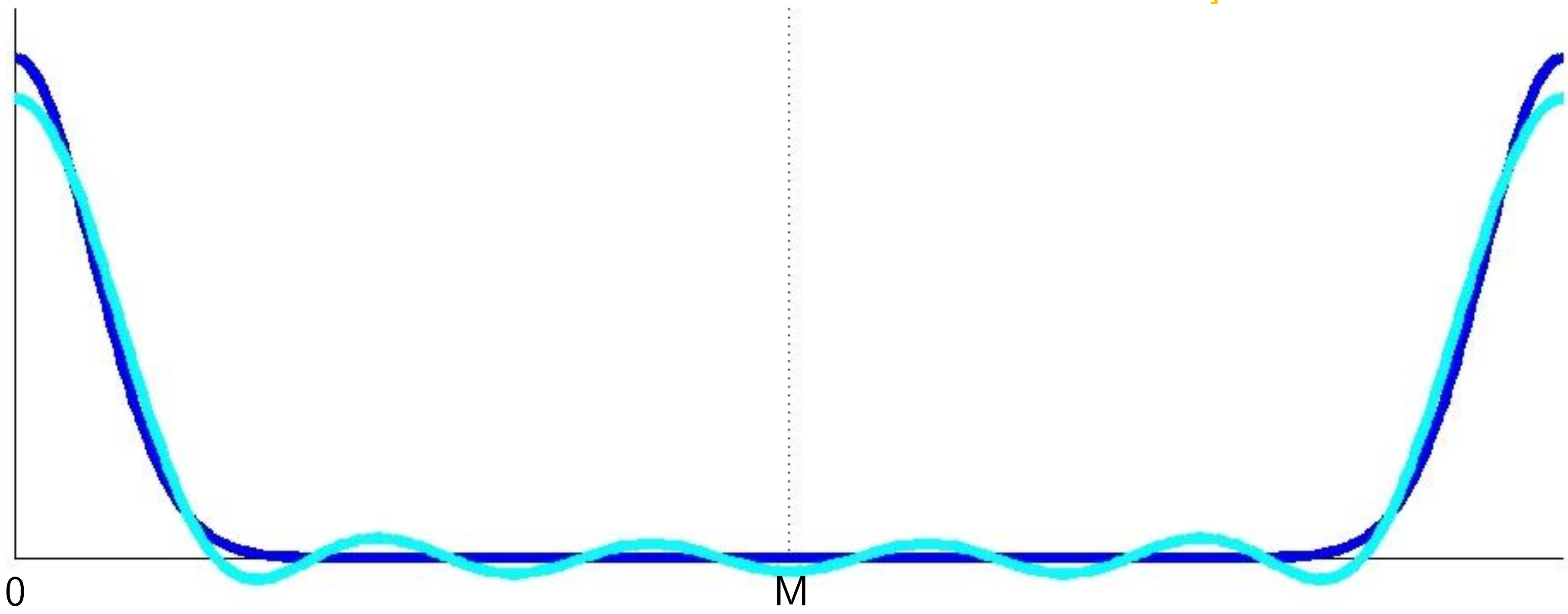
[Vedaldi & Zisserman '12]



Approximation using harmonic frequencies

Approximate a periodic function on interval [0, 2M] instead

[Vedaldi & Zisserman '12]

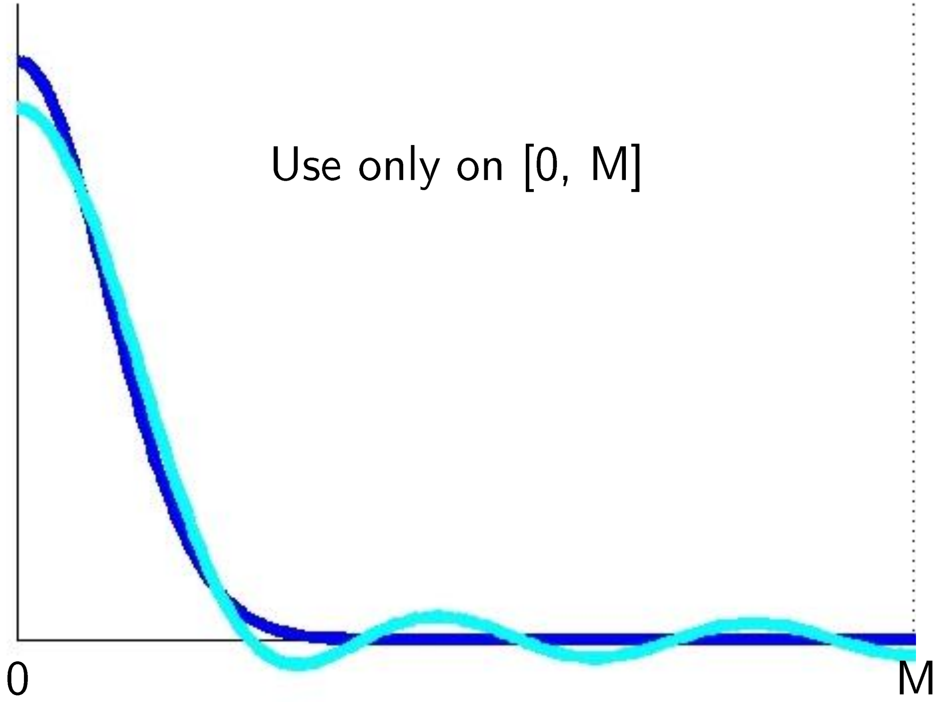


11D explicit feature map using harmonic frequencies

Approximation using harmonic frequencies

Approximate a periodic function on interval [0, 2M] instead

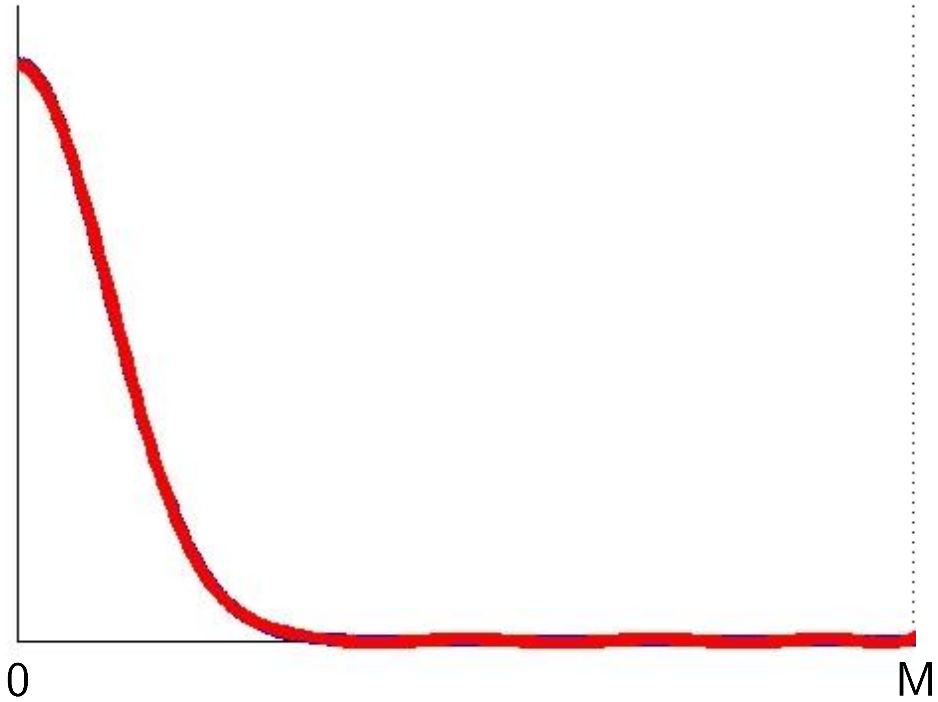
[Vedaldi & Zisserman ‘12]



11D explicit feature map using harmonic frequencies

Linear program approximation

Approximate on discrete subset of interval [0,M] using all frequencies [Chum '15]

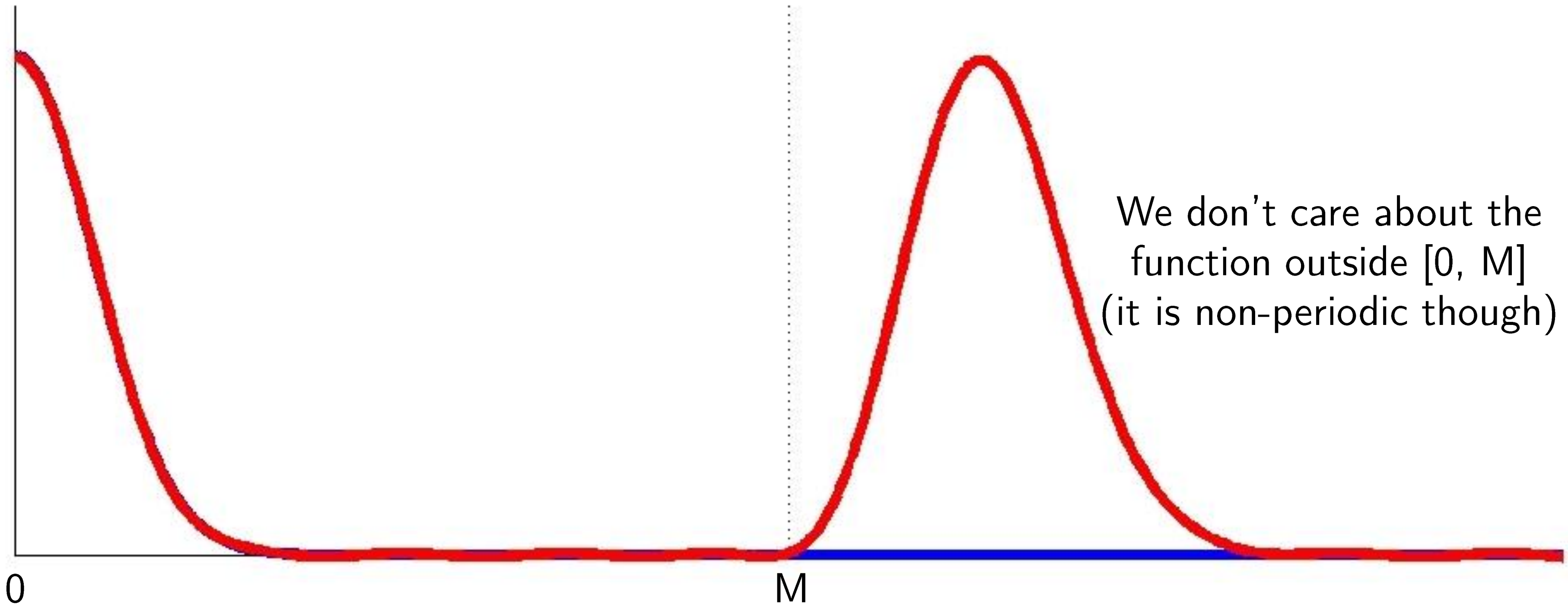


11D low dimensional explicit feature map

Linear program approximation

Approximate on discrete subset of interval [0,M] using all frequencies

[Chum '15]



We don't care about the function outside [0, M] (it is non-periodic though)

11D low dimensional explicit feature map

Low dimensional feature maps (LDFM)

[Chum ‘15]

Do not restrict to harmonic frequencies: $k(\lambda) \approx \sum_{\omega \in \Omega} \alpha_\omega \cos(\omega\lambda)$

LP relaxation: (discrete on [-M, M], discrete in frequencies)

minimize $$C_\infty(k, \hat{k}) = \max_{\boxed{z \in Z}} |k(z) - \hat{k}(z)|$$

Subject to $D(\hat{k}) = |\{\omega | \alpha_\omega > 0\}| = N$ relaxed to $\bar{D}(\hat{k}) = \sum_{\boxed{\omega \in \Omega}} \alpha_\omega = \bar{N}$

Equivalent to LP $$\min_{\hat{k}} \bar{D}(\hat{k}) + \gamma C(k, \hat{k})$$

Taylor approximation: (discrete on [-M, M], continuous in frequencies)

$$\hat{k}(\lambda) = \sum_{\omega \in \Omega} \alpha_\omega \cos((\omega + \boxed{d_\omega})\lambda) = \sum_{\omega \in \Omega} \alpha_\omega \cos(\omega\lambda) - \sum_{\omega \in \Omega} d_\omega \alpha_\omega \lambda \cos(\omega\lambda)$$

Efficient match kernels (EMK)

- Set representation and matching with EMK
- Joint encoding by modulation
- Applications of EMK

Efficient match kernels (EMK)

[Bo & Sminchisescu ‘09]

- Set representation
  - Image: set of local descriptors
  - Patch: set of pixels

- Set similarity by cross-matching all elements
  - Similarity for two elements: non-linear kernel

- EMK: linear similarity allows to aggregate per set

$$\mathrm{M}(\mathcal{P},\mathcal{Q}) = \sum_{p\in\mathcal{P}}\sum_{q\in\mathcal{Q}} k(p,q) \approx \sum_{p\in\mathcal{P}}\sum_{q\in\mathcal{Q}} \Psi(p)^\top\Psi(q) = \sum_{p\in\mathcal{P}}\Psi(p)^\top\sum_{q\in\mathcal{Q}}\Psi(q)$$

Efficient match kernels (EMK)

[Bo & Sminchisescu '09]

- Set representation
  - Image: set of local descriptors
  - Patch: set of pixels

- Set similarity by cross-matching all elements
  - Similarity for two elements: non-linear kernel

- EMK: linear similarity allows to aggregate per set

$$\mathrm{M}(\mathcal{P},\mathcal{Q}) = \sum_{\text{image 1}\ \boxed{p\in\mathcal{P}}} \sum_{q\in\mathcal{Q}} k(p,q) \approx \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}} \Psi(p)^\top \Psi(q) = \sum_{p\in\mathcal{P}} \Psi(p)^\top \sum_{q\in\mathcal{Q}} \Psi(q)$$

Efficient match kernels (EMK)

[Bo & Sminchisescu '09]

- Set representation
  - Image: set of local descriptors
  - Patch: set of pixels

- Set similarity by cross-matching all elements
  - Similarity for two elements: non-linear kernel

- EMK: linear similarity allows to aggregate per set

$$\mathrm{M}(\mathcal{P},\mathcal{Q}) = \sum_{\text{image 1}\ \boxed{p\in\mathcal{P}}} \sum_{\underset{\text{image 2}}{\boxed{q\in\mathcal{Q}}}} k(p,q) \approx \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}} \Psi(p)^\top \Psi(q) = \sum_{p\in\mathcal{P}} \Psi(p)^\top \sum_{q\in\mathcal{Q}} \Psi(q)$$

Efficient match kernels (EMK)

[Bo & Sminchisescu ‘09]

- Set representation
  - Image: set of local descriptors
  - Patch: set of pixels

- Set similarity by cross-matching all elements
  - Similarity for two elements: non-linear kernel

- EMK: linear similarity allows to aggregate per set

$$\mathrm{M}(\mathcal{P},\mathcal{Q}) = \sum_{\text{image 1}\ \boxed{p\in\mathcal{P}}} \sum_{\substack{\boxed{q\in\mathcal{Q}} \\ \text{image 2}}} \underset{\text{non-linear}}{\boxed{k(p,q)}} \approx \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}} \Psi(p)^\top \Psi(q) = \sum_{p\in\mathcal{P}} \Psi(p)^\top \sum_{q\in\mathcal{Q}} \Psi(q)$$

Efficient match kernels (EMK)

[Bo & Sminchisescu '09]

- Set representation
  - Image: set of local descriptors
  - Patch: set of pixels

- Set similarity by cross-matching all elements
  - Similarity for two elements: non-linear kernel

- EMK: linear similarity allows to aggregate per set

$$\mathrm{M}(\mathcal{P},\mathcal{Q}) = \sum_{\text{image 1}\,\boxed{p\in\mathcal{P}}} \sum_{\underset{\text{image 2}}{\boxed{q\in\mathcal{Q}}}} \underbrace{\boxed{k(p,q)}}_{\text{non-linear}} \approx \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}} \underbrace{\boxed{\Psi(p)^\top \Psi(q)}}_{\text{linear}} = \sum_{p\in\mathcal{P}} \Psi(p)^\top \sum_{q\in\mathcal{Q}} \Psi(q)$$

Efficient match kernels (EMK)

[Bo & Sminchisescu '09]

- Set representation
  - Image: set of local descriptors
  - Patch: set of pixels

- Set similarity by cross-matching all elements
  - Similarity for two elements: non-linear kernel

- EMK: linear similarity allows to aggregate per set

$$\mathrm{M}(\mathcal{P},\mathcal{Q}) = \sum_{\substack{\boxed{p\in\mathcal{P}} \\ \text{image 1}}} \sum_{\substack{\boxed{q\in\mathcal{Q}} \\ \text{image 2}}} \underbrace{\boxed{k(p,q)}}_{\text{non-linear}} \approx \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{Q}} \underbrace{\boxed{\Psi(p)^\top \Psi(q)}}_{\text{linear}} = \underbrace{\boxed{\sum_{p\in\mathcal{P}} \Psi(p)^\top}}_{\text{representation for image 1}} \underbrace{\boxed{\sum_{q\in\mathcal{Q}} \Psi(q)}}_{\text{representation for image 2}}$$

Kernel descriptors

Jointly encode multiple measurements by modulation

$$\Psi(p) = p_w \Psi(p_x) \otimes \Psi(p_y)$$

Joint similarity: product of different kernels

$$\Psi(p)^\top \Psi(q) = p_w q_w k(p_x, q_x) k(q_y, q_y)$$

Set representation: vector with $(2N_1+1)(2N_2+1)$ dimensions

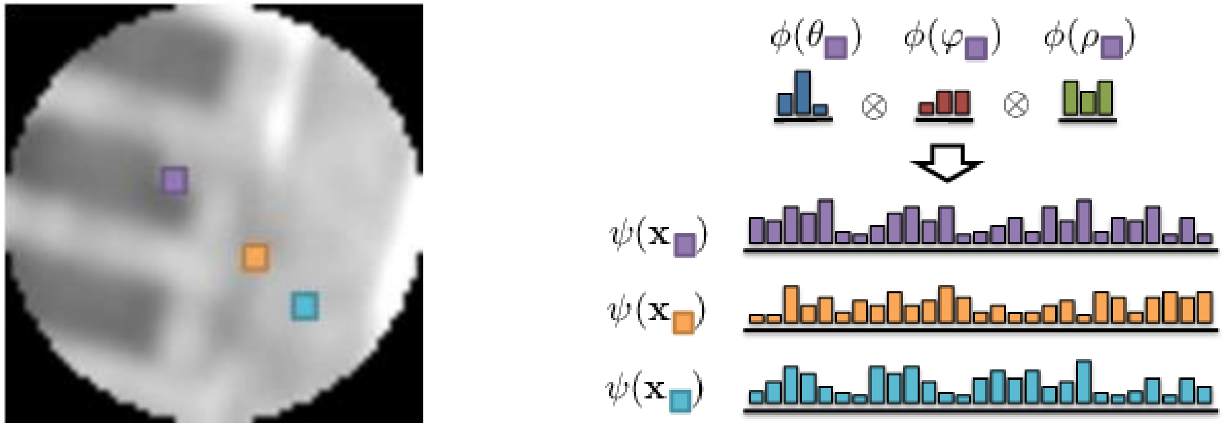
- aggregate representation for all elements

$$\mathcal{V}(\mathcal{P}) = \sum_{p \in \mathcal{P}} p_w \Psi(p_x) \otimes \Psi(p_y)$$

[Bo et al. '10]

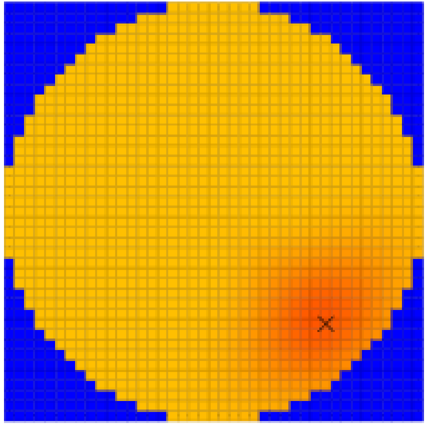
Local descriptors

Kernel descriptor – continuous encoding



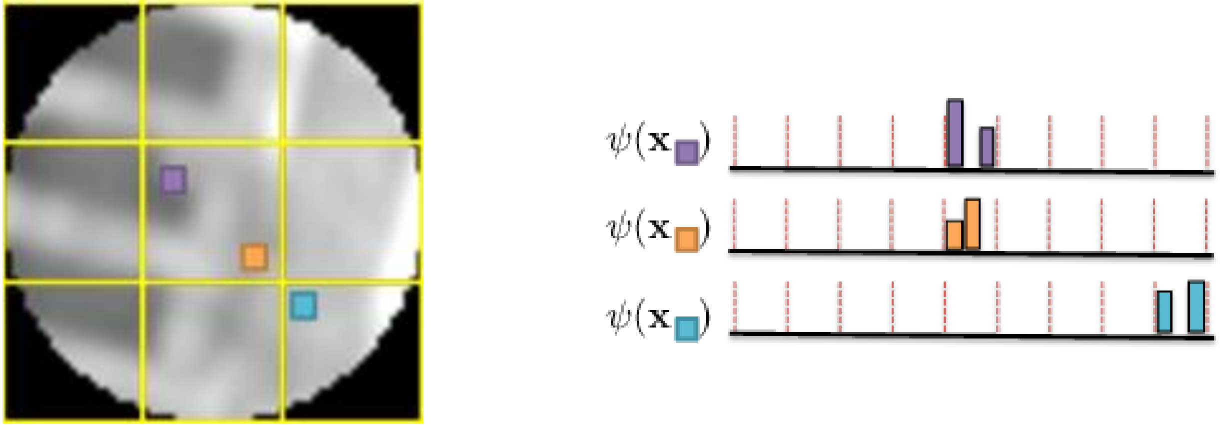
spatial similarity

red: high orange: low

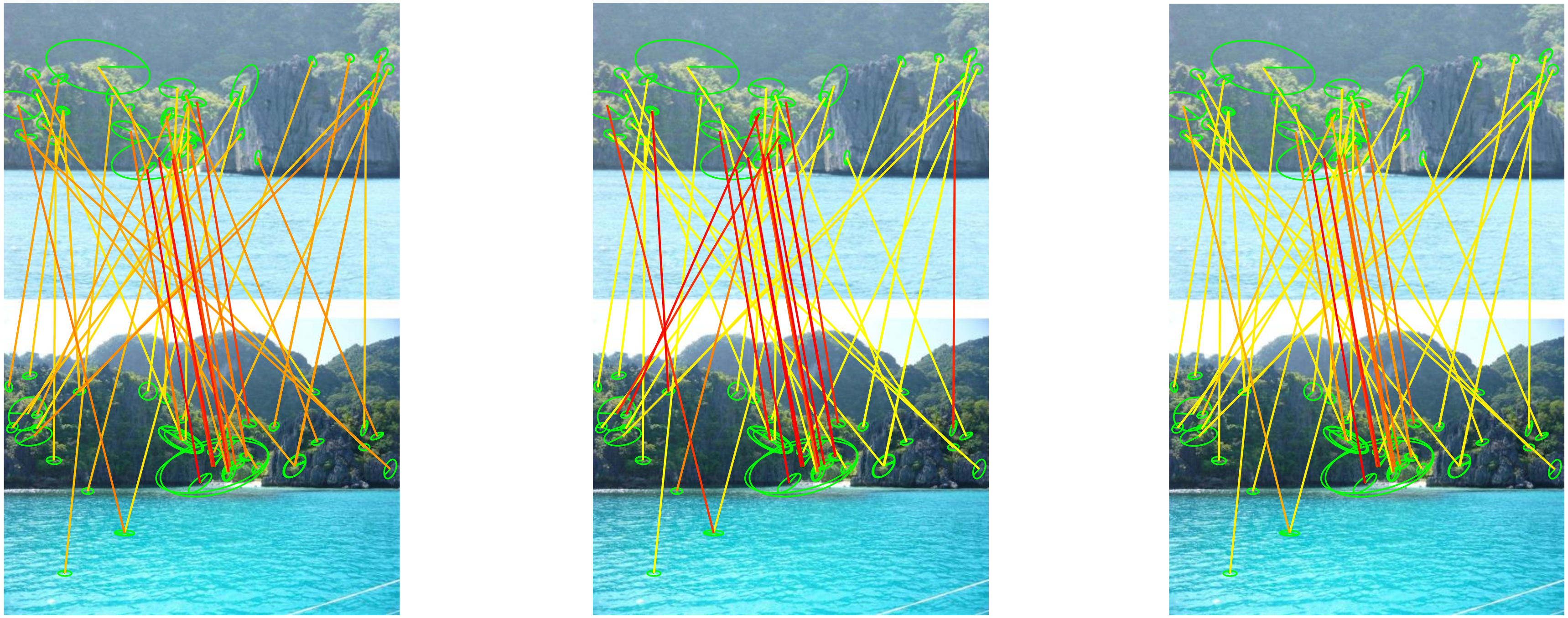


[Bursuc et al. '15]

SIFT-like – quantized encoding



Geometric image representation



only descriptor

only angle

both

red: high similarity yellow: low similarity

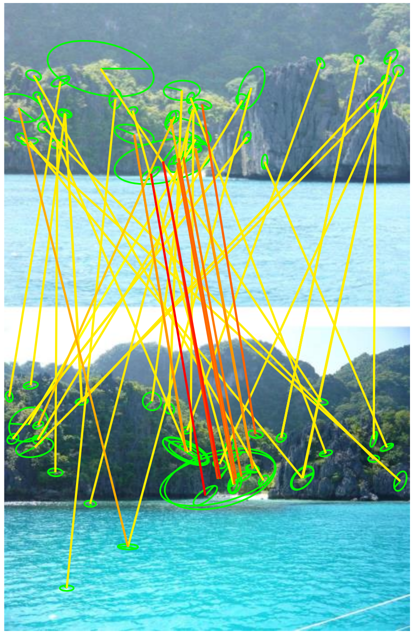
[Tolias et al. '15]

Trigonometric polynomial scores

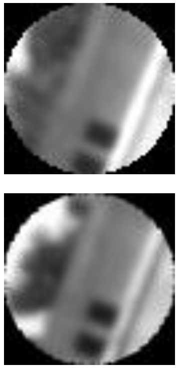
- 1D and 2D case
- Prior work on images and local patches

Transformation invariance

We assumed aligned objects

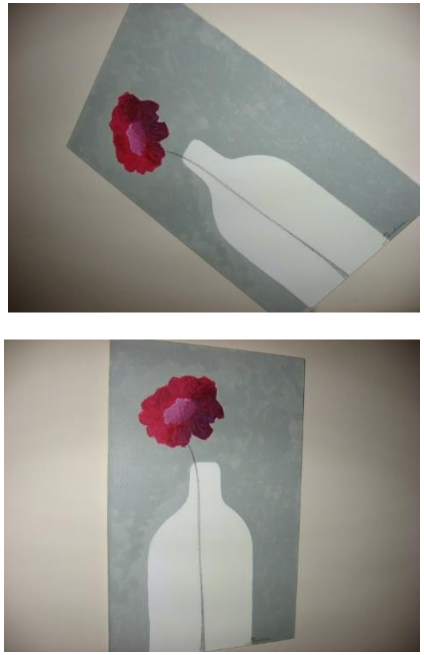


up-right images

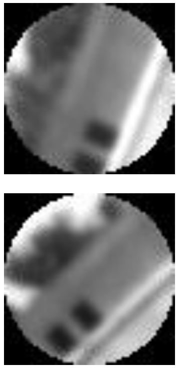


up-right patches

How to deal with transformations?



rotated images



rotated patches

Transformation invariance

- Shift by Δx

$$\cos(x - \Delta x) = \cos(x)\cos(\Delta x) + \sin(x)\sin(\Delta x)$$
$$\sin(x - \Delta x) = \sin(x)\cos(\Delta x) - \cos(x)\sin(\Delta x)$$

- Naive solution: construct shifted version by original descriptor
  - DM operations

- Similarity under shifting forms a trigonometric polynomial
  - 2D+(2N+1)M operations

$$\mathcal{S}(P_{\Delta_x}, Q) = \sum_{\omega \in \Omega_x} \left(\beta_\omega \cos(\omega \Delta_x) + \gamma_\omega \sin(\omega \Delta_x)\right)$$

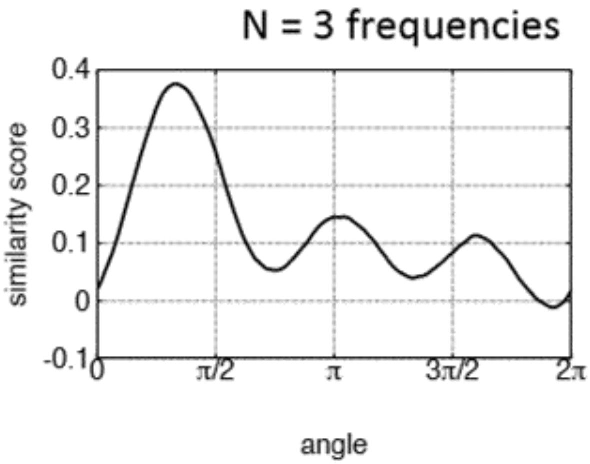
sub-vector inner products

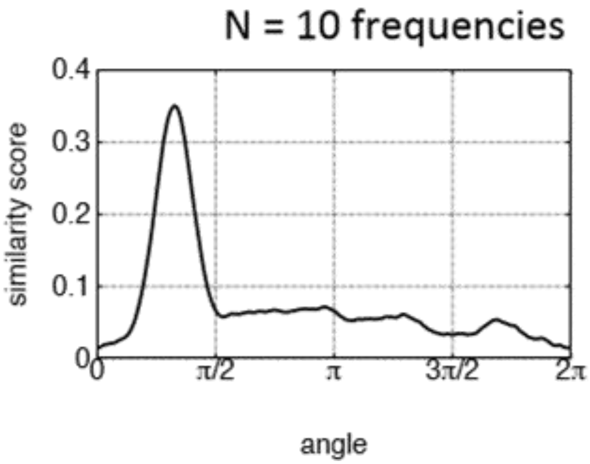
fixed grid of shifts

- Similarly for 2D shifts.

[Tolias et al. '15]

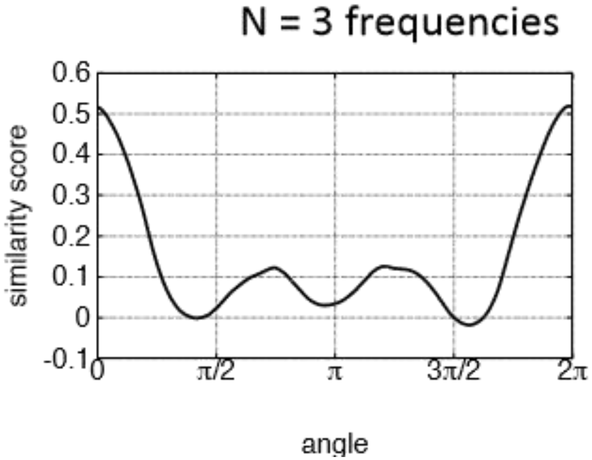
Rotation invariance

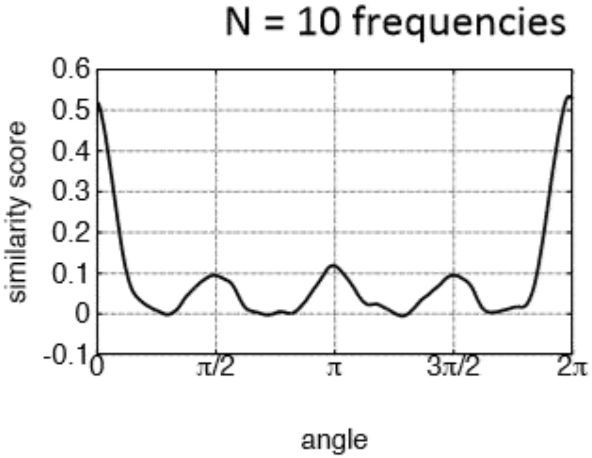




[Tolias et al. '15]

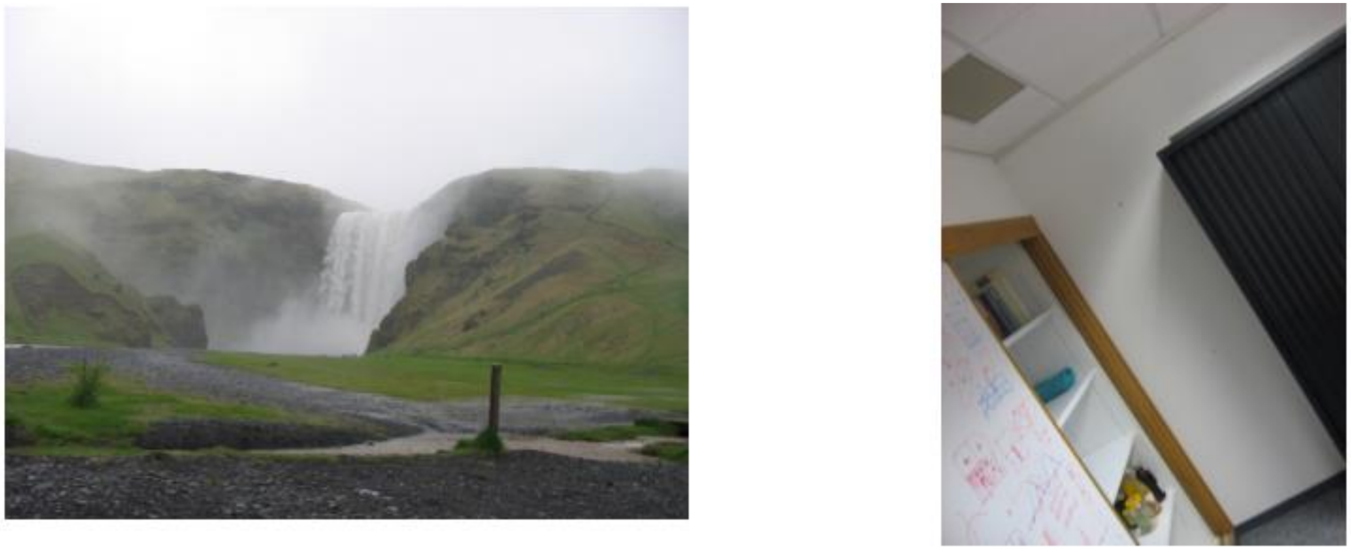
Rotation invariance

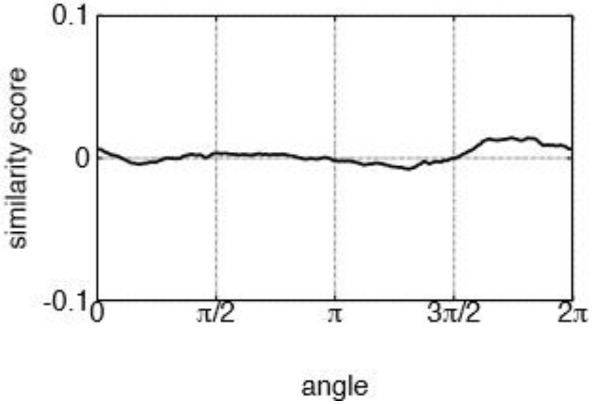

N = 3 frequencies

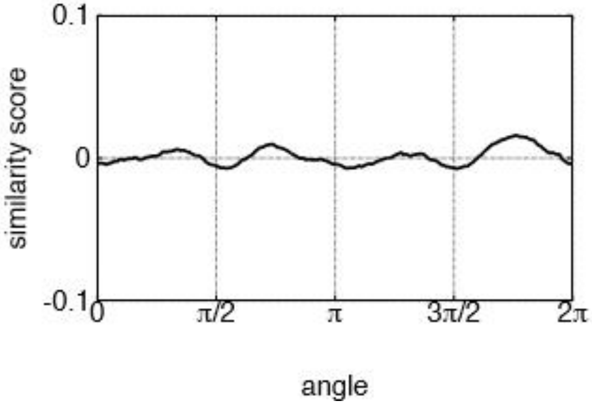

N = 10 frequencies

[Tolias et al. ‘15]

Rotation invariance

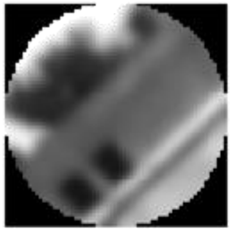



N = 3 frequencies

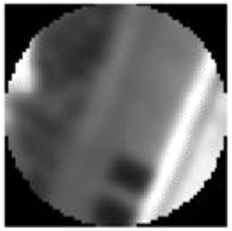

N = 10 frequencies

[Tolias et al. '15]

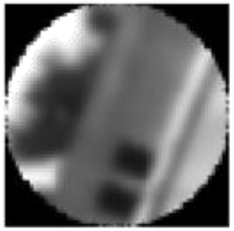
Efficient patch alignment



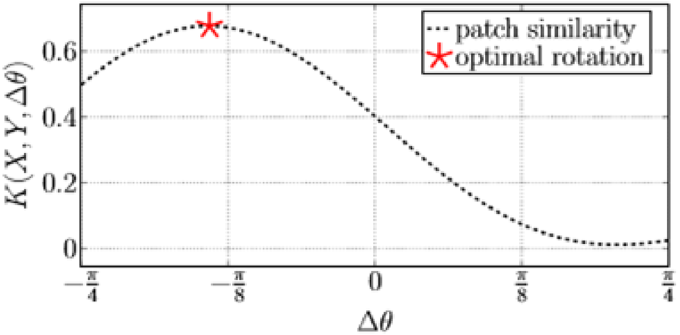
Patch A



Patch B



Aligned Patch A



Similarities for rotations of Patch A

[Bursuc et al. '15]

Asymmetric feature maps (AFM)

- Introduction of AFM
- Multiple kernel joint approximation
- 1D projections

Asymmetric feature maps (AFM)

$$k_1(q,p) \approx \sum_{\omega \in \Omega_1} \alpha_\omega \cos(\omega(q-p))$$

$$= \sum_{\omega \in \Omega_1} \underbrace{(\sqrt{\alpha_\omega}\cos(\omega q), \sqrt{\alpha_\omega}\sin(\omega q))}_{\Psi^1_\omega(q)}{}^\top \underbrace{(\sqrt{\alpha_\omega}\cos(\omega p), \sqrt{\alpha_\omega}\sin(\omega p))}_{\Psi^1_\omega(p)}$$

$$k_2(q,p) \approx \sum_{\omega \in \Omega_2} \alpha_\omega \cos(\omega(q-p))$$

$$= \sum_{\omega \in \Omega_2} \underbrace{(\sqrt{\alpha_\omega}\cos(\omega q), \sqrt{\alpha_\omega}\sin(\omega q))}_{\Psi^2_\omega(q)}{}^\top \underbrace{(\sqrt{\alpha_\omega}\cos(\omega p), \sqrt{\alpha_\omega}\sin(\omega p))}_{\Psi^2_\omega(p)}$$

Asymmetric feature maps (AFM)

$$\begin{aligned} k_1(q,p) &\approx \sum_{\omega\in\Omega_1} \alpha_\omega \cos(\omega(q-p)) \\ &= \sum_{\omega\in\Omega_1} \underbrace{(\cancel{\sqrt{\alpha_\omega}}\cos(\omega q), \cancel{\sqrt{\alpha_\omega}}\sin(\omega q))^\top}_{\Psi^1_\omega(q)} \underbrace{(\xcancel{\sqrt{\alpha_\omega}}\cos(\omega p), \xcancel{\sqrt{\alpha_\omega}}\sin(\omega p))}_{\Psi^1_\omega(p)} \end{aligned}$$

$$\begin{aligned} k_2(q,p) &\approx \sum_{\omega\in\Omega_2} \alpha_\omega \cos(\omega(q-p)) \\ &= \sum_{\omega\in\Omega_2} \underbrace{(\sqrt{\alpha_\omega}\cos(\omega q), \sqrt{\alpha_\omega}\sin(\omega q))^\top}_{\Psi^2_\omega(q)} \underbrace{(\sqrt{\alpha_\omega}\cos(\omega p), \sqrt{\alpha_\omega}\sin(\omega p))}_{\Psi^2_\omega(p)} \end{aligned}$$

Asymmetric feature maps (AFM)

$$
\begin{aligned}
k_1(q,p) &\approx \sum_{\omega\in\Omega_1} \alpha_\omega \cos(\omega(q-p)) \\
&= \sum_{\omega\in\Omega_1} \underbrace{(\cancel{\sqrt{\alpha_\omega}}\cos(\omega q), \cancel{\sqrt{\alpha_\omega}}\sin(\omega q))^\top}_{\Psi^1_\omega(q)} \underbrace{(\xcancel{\sqrt{\alpha_\omega}}\cos(\omega p), \xcancel{\sqrt{\alpha_\omega}}\sin(\omega p))}_{\Psi^1_\omega(p)}
\end{aligned}
$$

$$
\begin{aligned}
k_2(q,p) &\approx \sum_{\omega\in\Omega_2} \alpha_\omega \cos(\omega(q-p)) \\
&= \sum_{\omega\in\Omega_2} \underbrace{(\cancel{\sqrt{\alpha_\omega}}\cos(\omega q), \cancel{\sqrt{\alpha_\omega}}\sin(\omega q))^\top}_{\Psi^2_\omega(q)} \underbrace{(\xcancel{\sqrt{\alpha_\omega}}\cos(\omega p), \xcancel{\sqrt{\alpha_\omega}}\sin(\omega p))}_{\Psi^2_\omega(p)}
\end{aligned}
$$

Asymmetric feature maps (AFM)

$$k_1(q,p) \approx \sum_{\omega\in\Omega_1} \alpha_\omega \cos(\omega(q-p))$$

$$= \sum_{\omega\in\Omega_1} \underbrace{(\ \alpha_\omega \cos(\omega q),\ \ \alpha_\omega \sin(\omega q))^\top}_{\Psi^1_\omega(q)} \underbrace{(\ \ \cos(\omega p),\ \ \ \sin(\omega p))}_{\Psi^1_\omega(p)}$$

$$k_2(q,p) \approx \sum_{\omega\in\Omega_2} \alpha_\omega \cos(\omega(q-p))$$

$$= \sum_{\omega\in\Omega_2} \underbrace{(\ \alpha_\omega \cos(\omega q),\ \ \alpha_\omega \sin(\omega q))^\top}_{\Psi^2_\omega(q)} \underbrace{(\ \ \cos(\omega p),\ \ \ \sin(\omega p))}_{\Psi^2_\omega(p)}$$

Asymmetric feature maps (AFM)

$$k_1(q,p) \approx \sum_{\omega\in\Omega_{\xcancel{1}}} \alpha_\omega \cos(\omega(q-p))$$

$$= \sum_{\omega\in\Omega_{\xcancel{1}}} (\underbrace{\alpha_\omega \cos(\omega q), \quad \alpha_\omega \sin(\omega q))^\top}_{\Psi^1_\omega(q)} (\underbrace{\cos(\omega p), \quad \sin(\omega p))}_{\Psi^1_\omega(p)}$$

$$k_2(q,p) \approx \sum_{\omega\in\Omega_{\xcancel{2}}} \alpha_\omega \cos(\omega(q-p))$$

$$= \sum_{\omega\in\Omega_{\xcancel{2}}} (\underbrace{\alpha_\omega \cos(\omega q), \quad \alpha_\omega \sin(\omega q))^\top}_{\Psi^2_\omega(q)} (\underbrace{\cos(\omega p), \quad \sin(\omega p))}_{\Psi^2_\omega(p)}$$

Asymmetric feature maps (AFM)

$$k_1(q,p) \approx \sum_{\omega\in\Omega} \alpha_\omega \cos(\omega(q-p))$$

$$= \sum_{\omega\in\Omega} \underbrace{\left(\alpha_\omega \cos(\omega q), \quad \alpha_\omega \sin(\omega q)\right)}_{\Psi^1_\omega(q)}{}^\top \underbrace{\left(\quad \cos(\omega p), \quad \sin(\omega p)\right)}_{\Psi^{\cancel{1}}_\omega(p)}$$

$$k_2(q,p) \approx \sum_{\omega\in\Omega} \alpha_\omega \cos(\omega(q-p))$$

$$= \sum_{\omega\in\Omega} \underbrace{\left(\alpha_\omega \cos(\omega q), \quad \alpha_\omega \sin(\omega q)\right)}_{\Psi^2_\omega(q)}{}^\top \underbrace{\left(\quad \cos(\omega p), \quad \sin(\omega p)\right)}_{\Psi^{\cancel{2}}_\omega(p)}$$

Asymmetric feature maps (AFM)

$$
\begin{aligned}
k_1(q,p) &\approx \sum_{\omega\in\Omega} \alpha_\omega \cos(\omega(q-p)) \\
&= \sum_{\omega\in\Omega} \underbrace{(\ \alpha_\omega \cos(\omega q),\quad \alpha_\omega \sin(\omega q))^\top}_{\Psi^1_\omega(q)} \underbrace{(\quad \cos(\omega p),\quad \sin(\omega p))}_{\Psi_\omega(p)}
\end{aligned}
$$

$$
\begin{aligned}
k_2(q,p) &\approx \sum_{\omega\in\Omega} \alpha_\omega \cos(\omega(q-p)) \\
&= \sum_{\omega\in\Omega} \underbrace{(\ \alpha_\omega \cos(\omega q),\quad \alpha_\omega \sin(\omega q))^\top}_{\Psi^2_\omega(q)} \underbrace{(\quad \cos(\omega p),\quad \sin(\omega p))}_{\Psi_\omega(p)}
\end{aligned}
$$

Asymmetric feature maps (AFM)

$$
\begin{aligned}
k_1(q,p) &\approx \sum_{\omega\in\Omega} \alpha_\omega \cos(\omega(q-p)) \\
&= \sum_{\omega\in\Omega} (\underbrace{\alpha_\omega \cos(\omega q), \quad \alpha_\omega \sin(\omega q))^\top}_{\Psi^1_\omega(q)} \underbrace{(\quad \cos(\omega p), \quad \sin(\omega p))}_{\Psi_\omega(p)}
\end{aligned}
$$

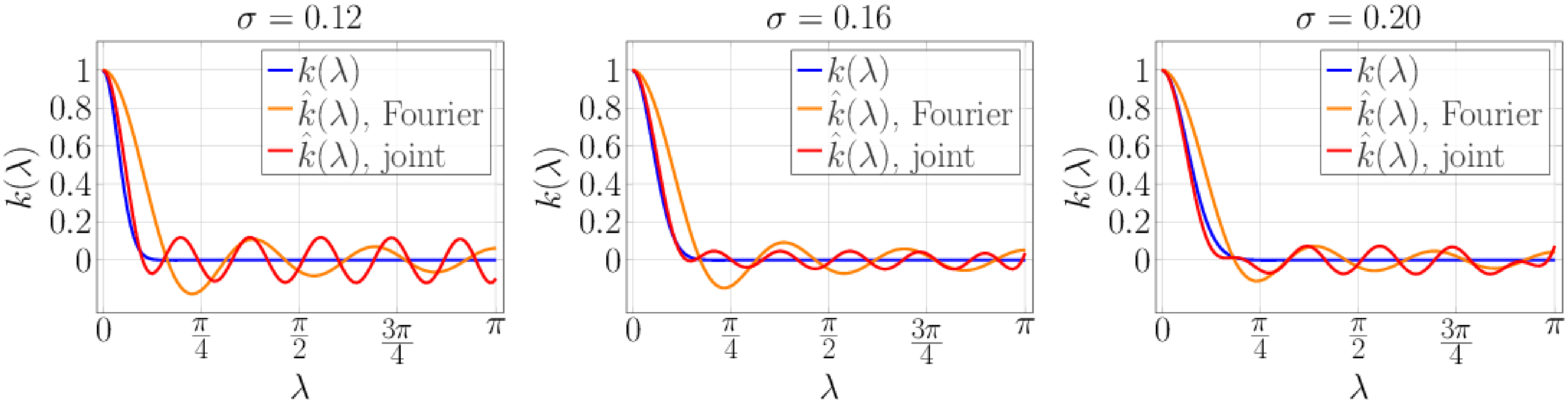
$$
\begin{aligned}
k_2(q,p) &\approx \sum_{\omega\in\Omega} \alpha_\omega \cos(\omega(q-p)) \\
&= \sum_{\omega\in\Omega} (\underbrace{\alpha_\omega \cos(\omega q), \quad \alpha_\omega \sin(\omega q))^\top}_{\Psi^2_\omega(q)} \underbrace{(\quad \cos(\omega p), \quad \sin(\omega p))}_{\Psi_\omega(p)}
\end{aligned}
$$

- One side defines the kernel
- One descriptor for multiple similarity measures
- Savings in memory

Joint kernel approximation

- AFM compatible with Harmonic frequencies, but not with LDFM

- We jointly optimize LDFM for multiple kernels



Sketch-based image retrieval

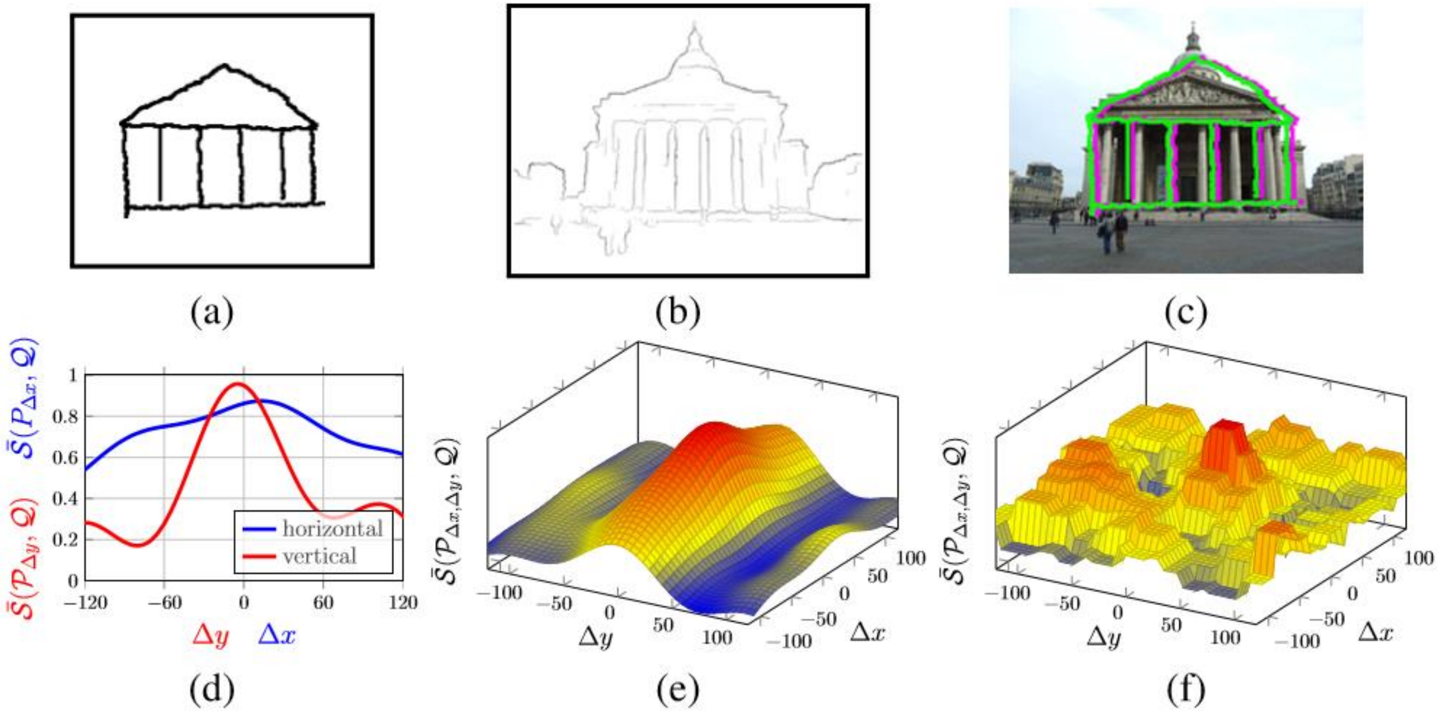
- Sketch descriptor
- Translation and scale invariance
- Ranking and re-ranking procedures

Sketch descriptor

- Edge detection: (edge strength, gradient angle) per pixel [Dollar & Zitnick '13]
- Kernel descriptor encodes position (x,y), angle and strength

$$\mathcal{V}(\mathcal{P}) = \sum_{p \in \mathcal{P}} p_w \Psi_{xy}(p_x) \otimes \Psi_{xy}(p_y) \otimes \Psi_o(p_o)$$

Translation invariance



(c): Alignment with 1D projections (magenta) and with full 2D (green)

(d): 1D translations

(e): Full 2D trigonometric polynomial

(f): Full 2D with binary coefficients/variables

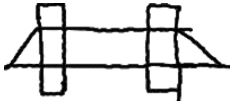
Scale invariance

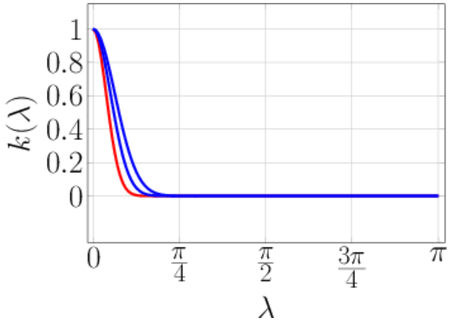
Query

Kernel used

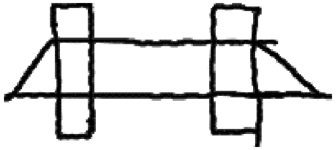
Result

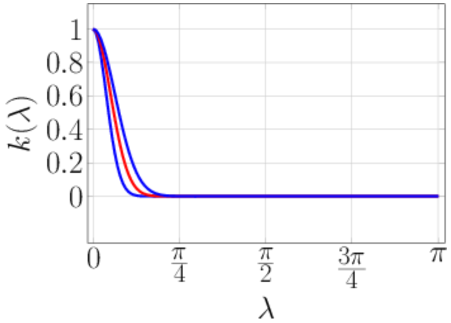
Scale 1



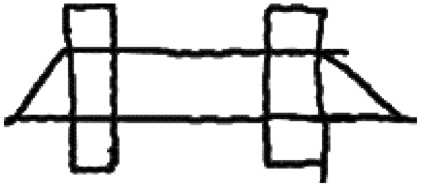


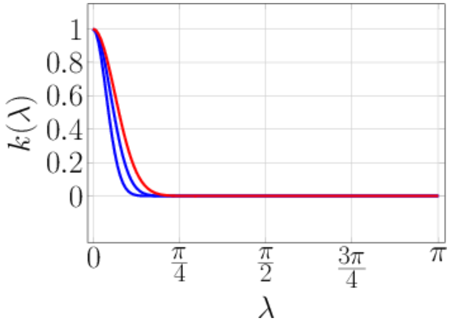
Scale 2

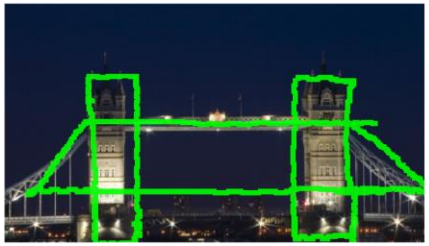




Scale 3

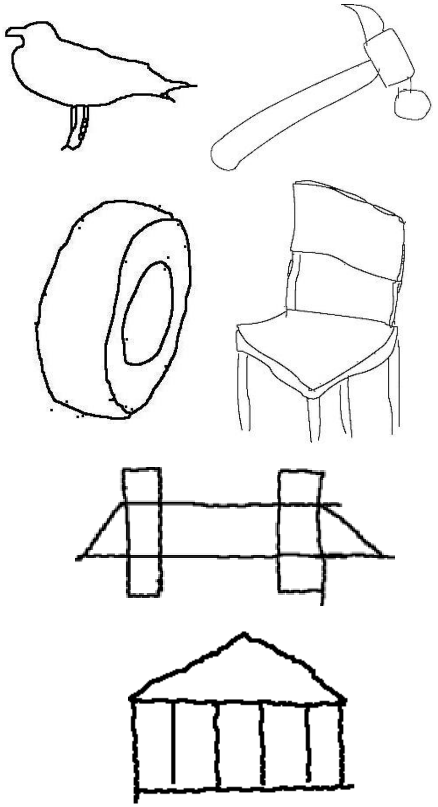






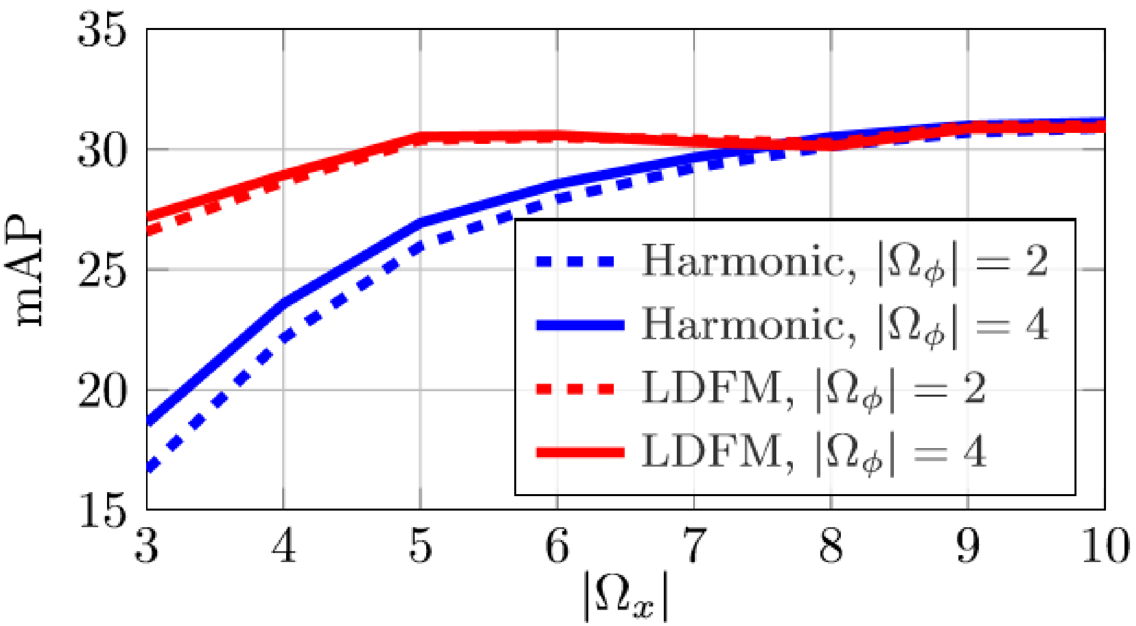
Sketch-based image retrieval

- Initial ranking (fast)
  - 1D projections on x and y: sum of scores
  - Use the discriminative projection first
- Re-ranking (precise)
  - Input from 1D projections – locally refine
  - Full 2D with binary approximation

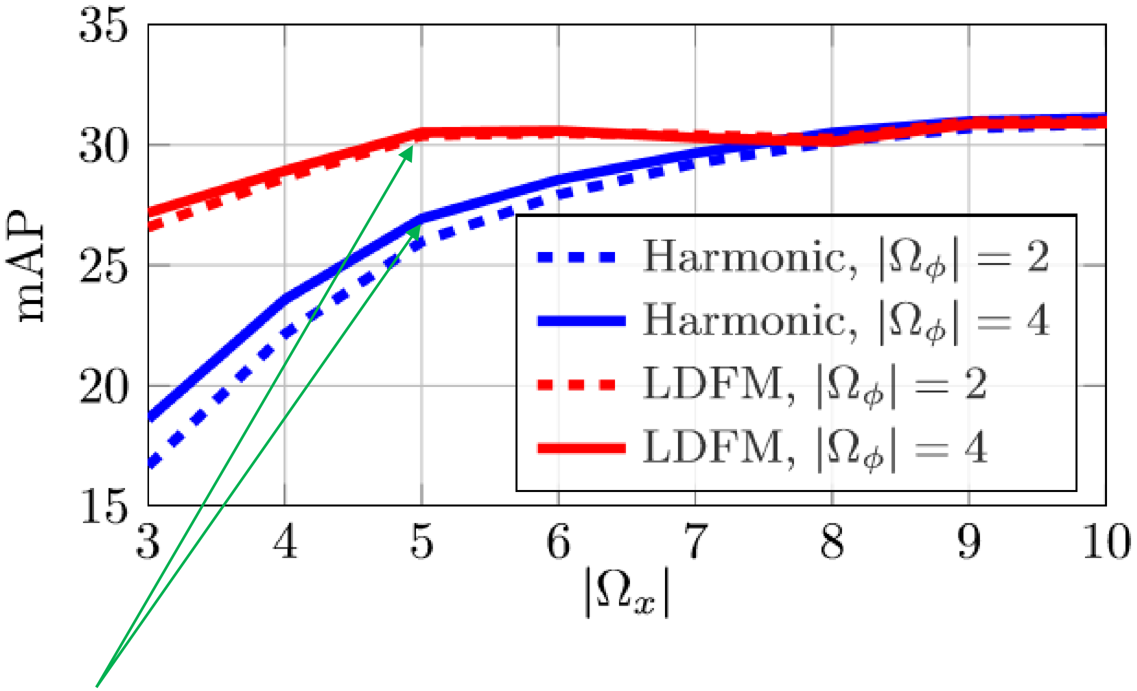


Results on sketch-based image retrieval

Joint LDFM approximation

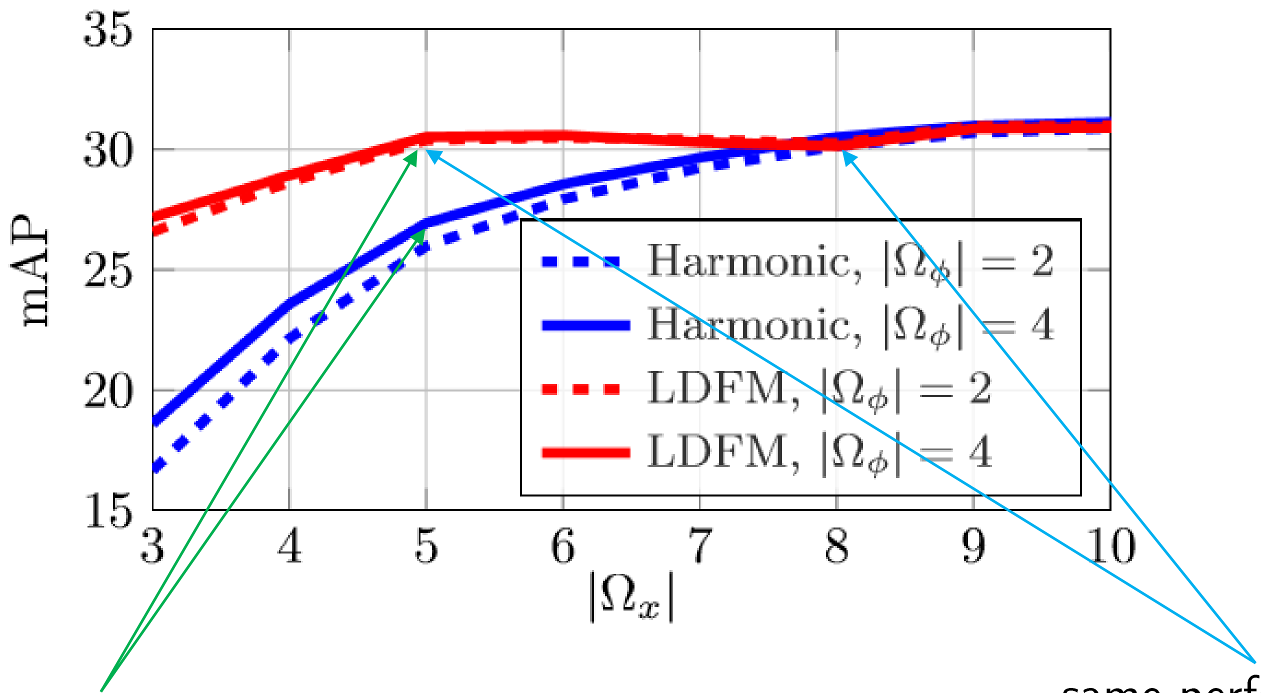


Joint LDFM approximation



243D vector per image
27D for 1D projections

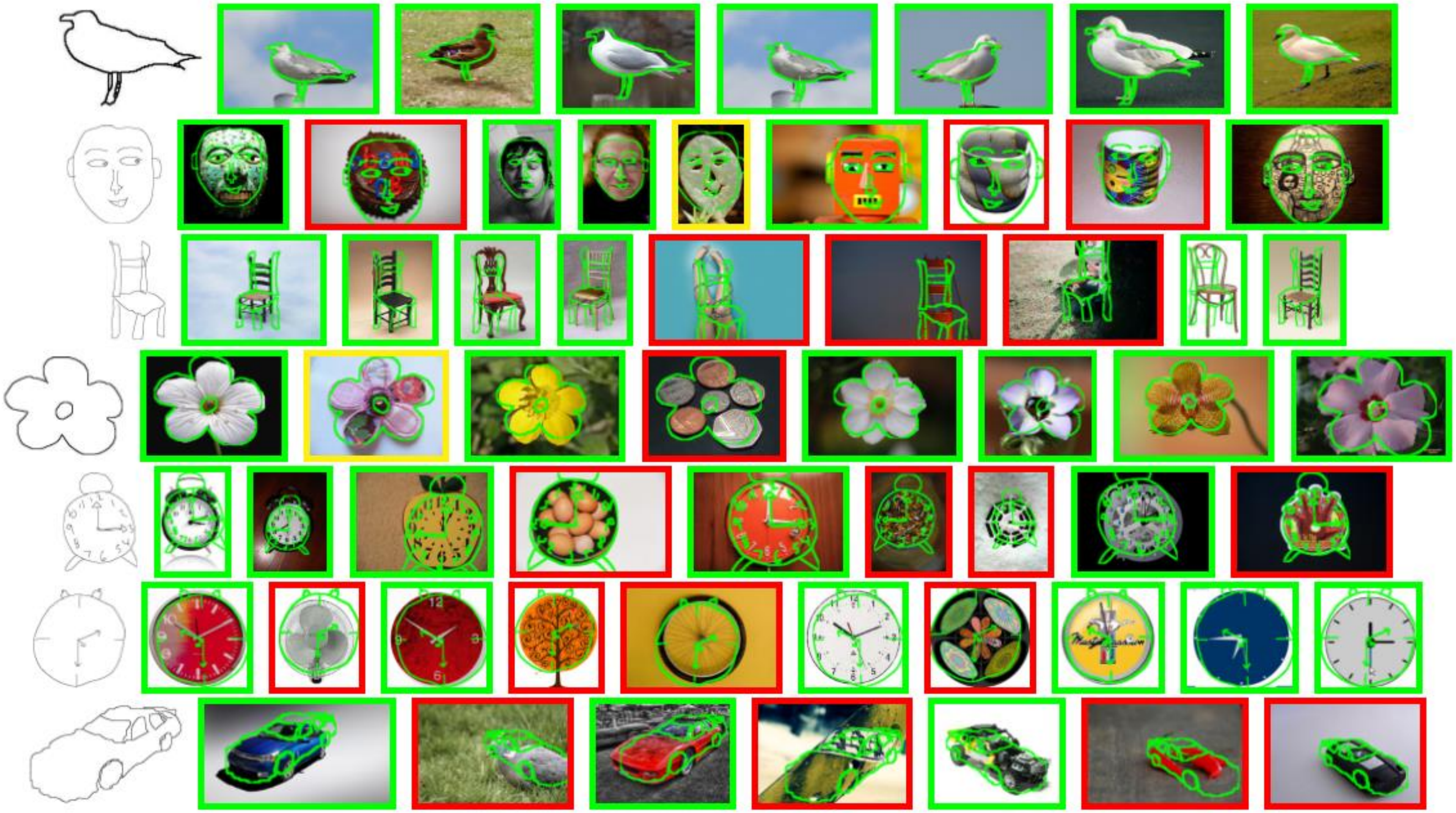
Joint LDFM approximation



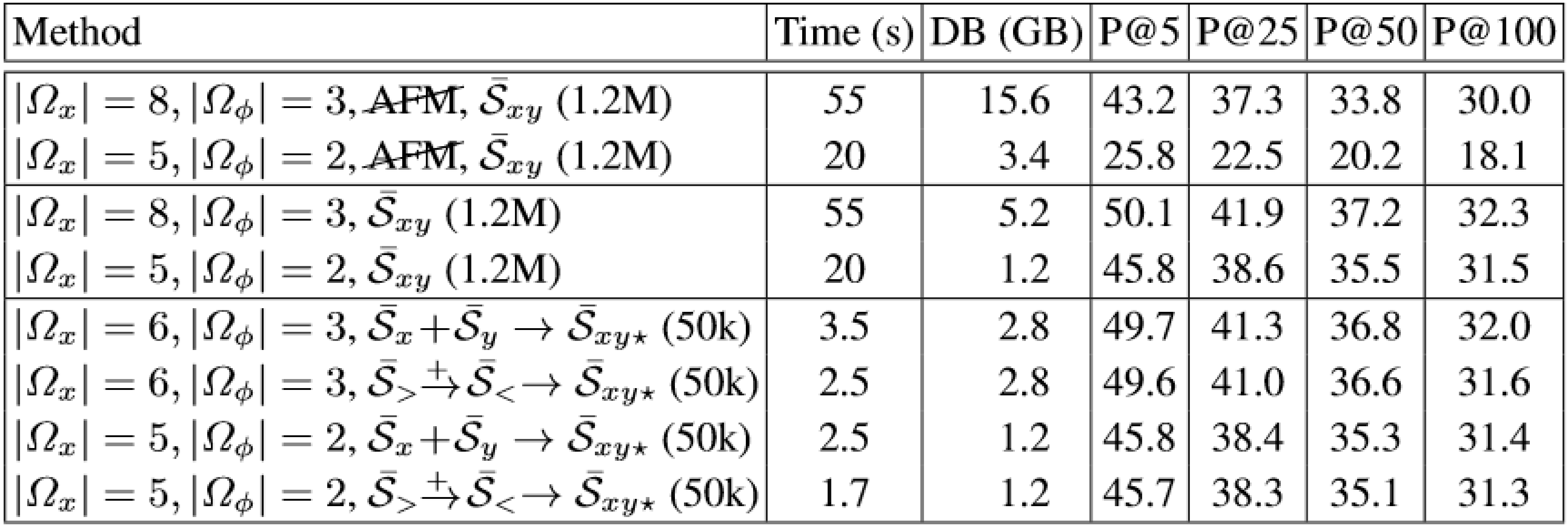
243D vector per image
27D for 1D projections

same performance
243D for LDFM
675D for Harmonic

Retrieval examples



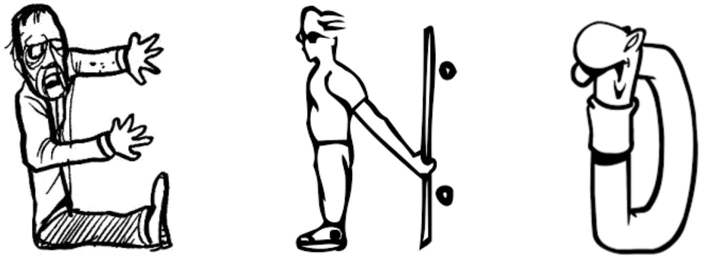
Evaluation on 1.2M images



Method	Time (s)	DB (GB)	P@5	P@25	P@50	P@100
$\lvert\Omega_x\rvert = 8, \lvert\Omega_\phi\rvert = 3,$ ~~AFM~~, $\bar{\mathcal{S}}_{xy}$ (1.2M)	55	15.6	43.2	37.3	33.8	30.0
$\lvert\Omega_x\rvert = 5, \lvert\Omega_\phi\rvert = 2,$ ~~AFM~~, $\bar{\mathcal{S}}_{xy}$ (1.2M)	20	3.4	25.8	22.5	20.2	18.1
$\lvert\Omega_x\rvert = 8, \lvert\Omega_\phi\rvert = 3, \bar{\mathcal{S}}_{xy}$ (1.2M)	55	5.2	50.1	41.9	37.2	32.3
$\lvert\Omega_x\rvert = 5, \lvert\Omega_\phi\rvert = 2, \bar{\mathcal{S}}_{xy}$ (1.2M)	20	1.2	45.8	38.6	35.5	31.5
$\lvert\Omega_x\rvert = 6, \lvert\Omega_\phi\rvert = 3, \bar{\mathcal{S}}_x + \bar{\mathcal{S}}_y \to \bar{\mathcal{S}}_{xy\star}$ (50k)	3.5	2.8	49.7	41.3	36.8	32.0
$\lvert\Omega_x\rvert = 6, \lvert\Omega_\phi\rvert = 3, \bar{\mathcal{S}}_> \xrightarrow{+} \bar{\mathcal{S}}_< \to \bar{\mathcal{S}}_{xy\star}$ (50k)	2.5	2.8	49.6	41.0	36.6	31.6
$\lvert\Omega_x\rvert = 5, \lvert\Omega_\phi\rvert = 2, \bar{\mathcal{S}}_x + \bar{\mathcal{S}}_y \to \bar{\mathcal{S}}_{xy\star}$ (50k)	2.5	1.2	45.8	38.4	35.3	31.4
$\lvert\Omega_x\rvert = 5, \lvert\Omega_\phi\rvert = 2, \bar{\mathcal{S}}_> \xrightarrow{+} \bar{\mathcal{S}}_< \to \bar{\mathcal{S}}_{xy\star}$ (50k)	1.7	1.2	45.7	38.3	35.1	31.3

Conclusions

- Achievements compared to existing kernel approaches
  - x10 speedup compared to full 2D trig. pol. on all images
  - 3 times less memory due to AFM
  - Better performance due to joint kernel approximation


END