Multiple-Kernel Local-Patch Descriptor

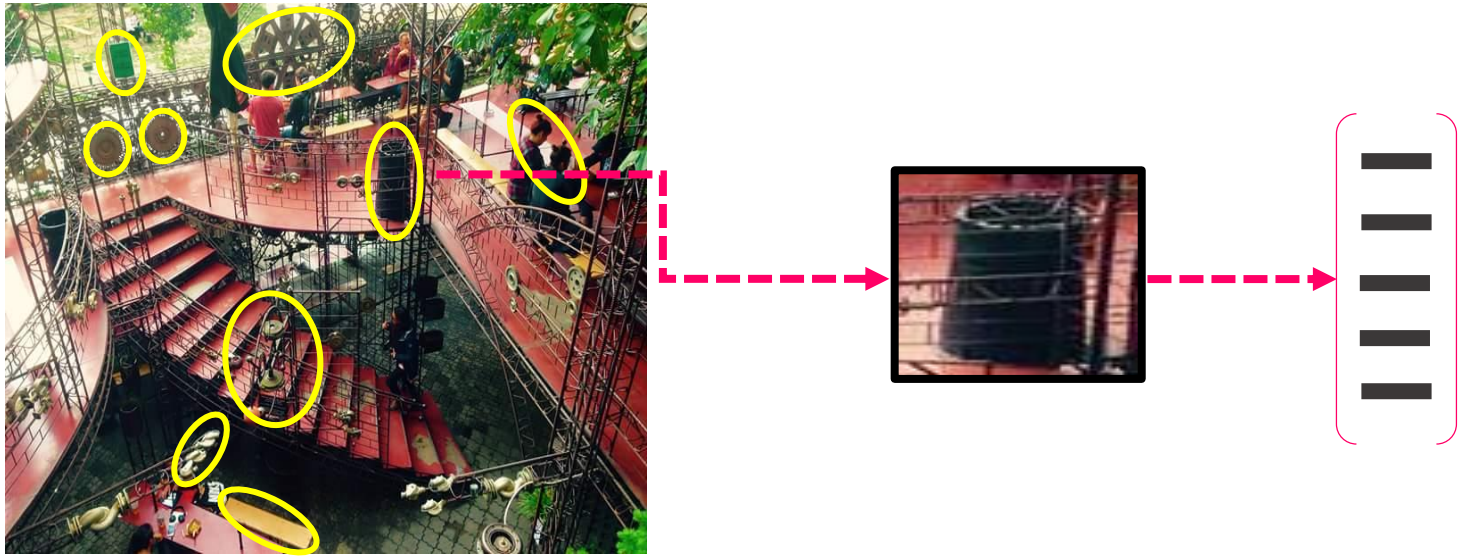
A. Mukundan G. Tolias O. Chum

Visual Recognition Group (VRG),
Czech Technical University in Prague

BMVC 2017

Local-patch descriptors

- Detect repeatable keypoints
- Normalize to square patches
- Extract local-patch descriptor



Tasks:

- Structure-from-Motion,
- Multi-View Stereo,
- Image-based localization,
- Image retrieval

Related work

Hand-crafted descriptors

- SIFT, SURF, Daisy, BRIEF, ORB, BRISK, ...
- Intuitive
- Tweakable
- No learning
- Low performance

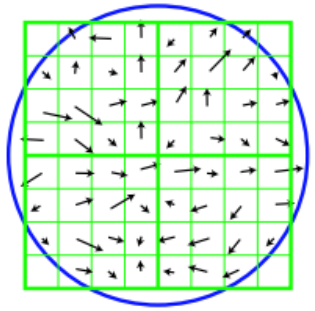


Figure from [Lowe, IJCV '04]

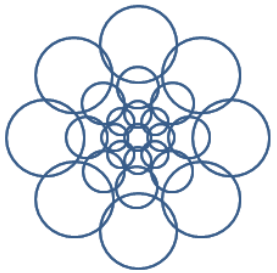


Figure from [Tola, et al, PAMI '10]

Learned descriptors

- Dominated by CNN-based: DDesc, TFeat, DeepCompare, ...
- Lots of training data
- Long training time
- Behavior difficult to interpret
- High performance

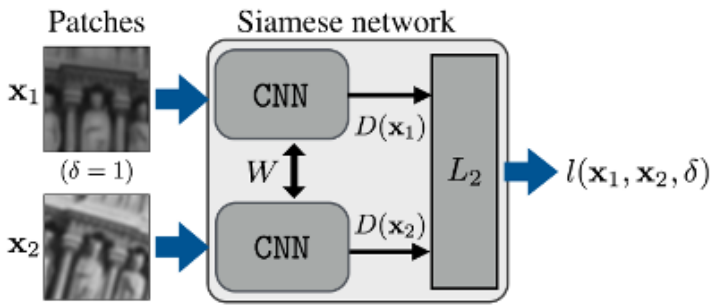


Figure from [Simo-Serra, et al ICCV '15]

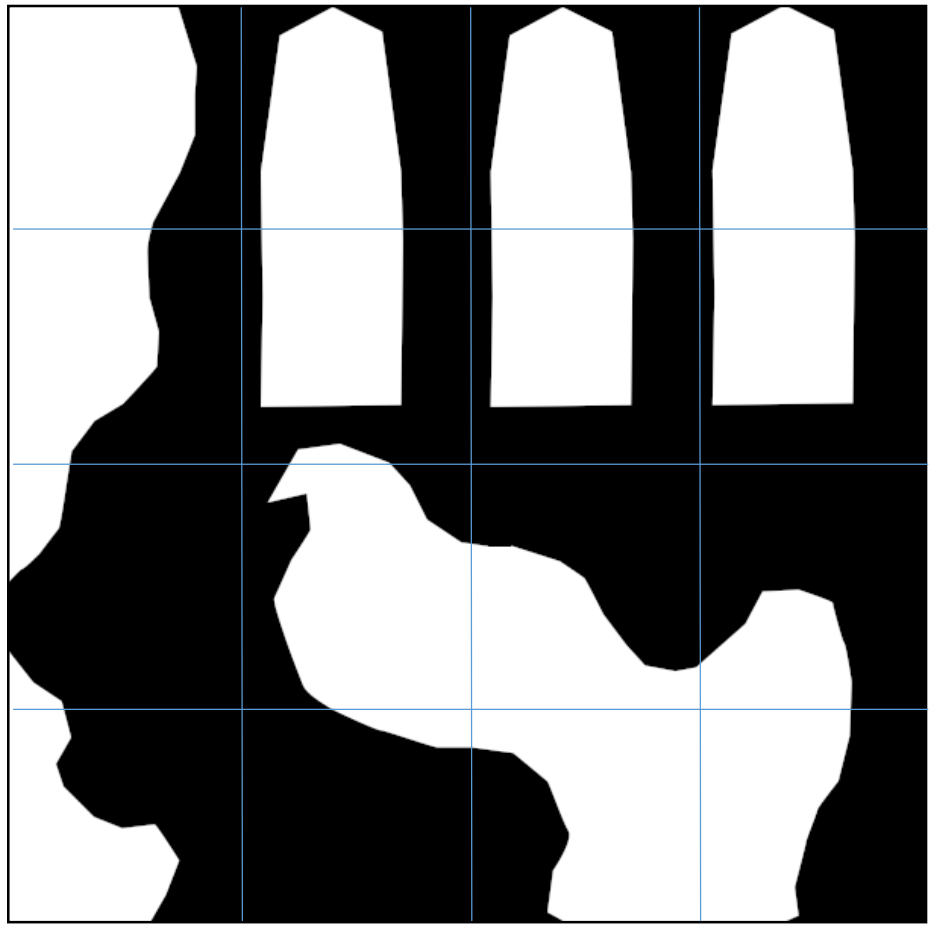
Overview

- Hand-crafted design → kernel descriptor
- Inject supervision → data whitening from matching pairs of patches

- Intuitive design
- Understanding of learned patch similarity
- Performance boosted by supervision

A familiar descriptor : SIFT

Local patch divided into 4 x 4 spatial bins

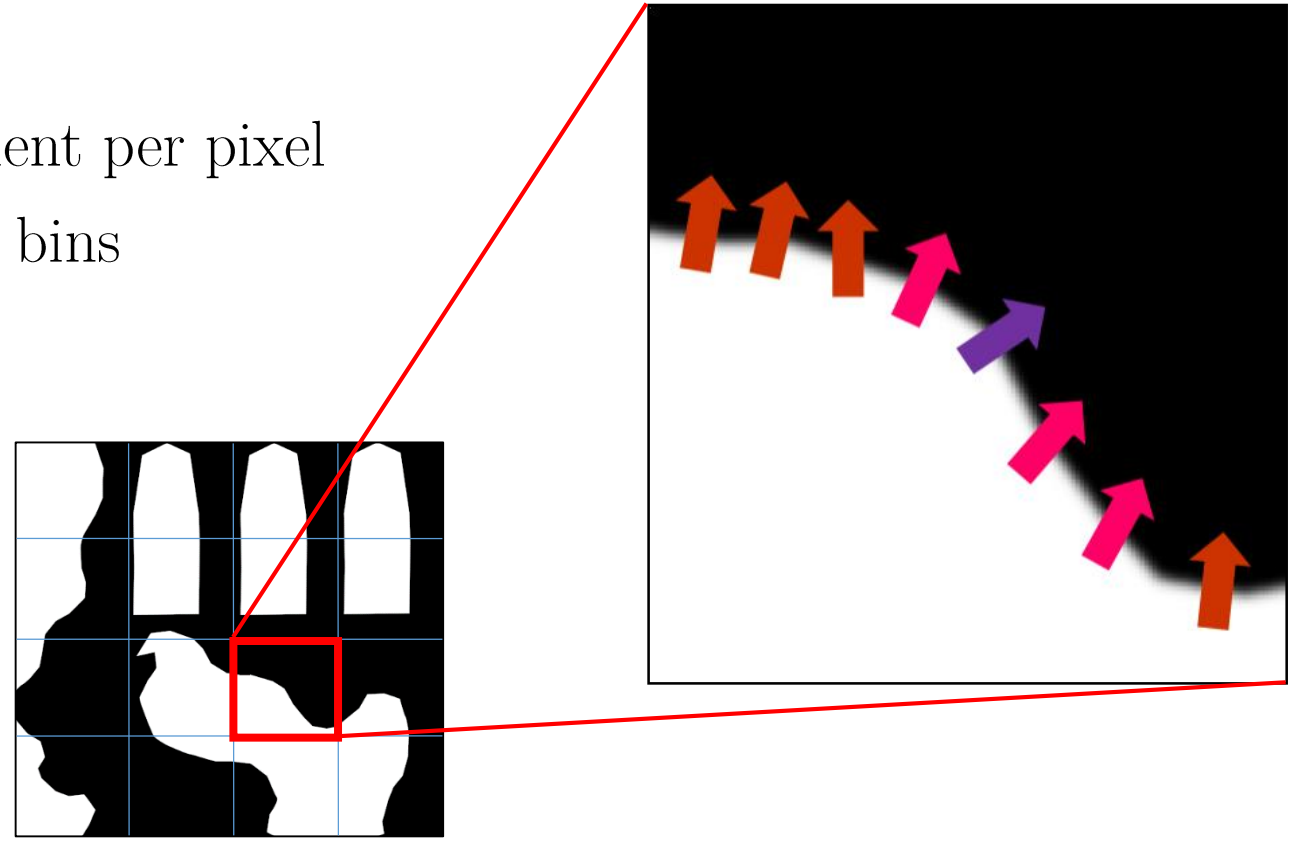


[D. Lowe, IJCV '04]

SIFT (continued)

In each spatial bin:

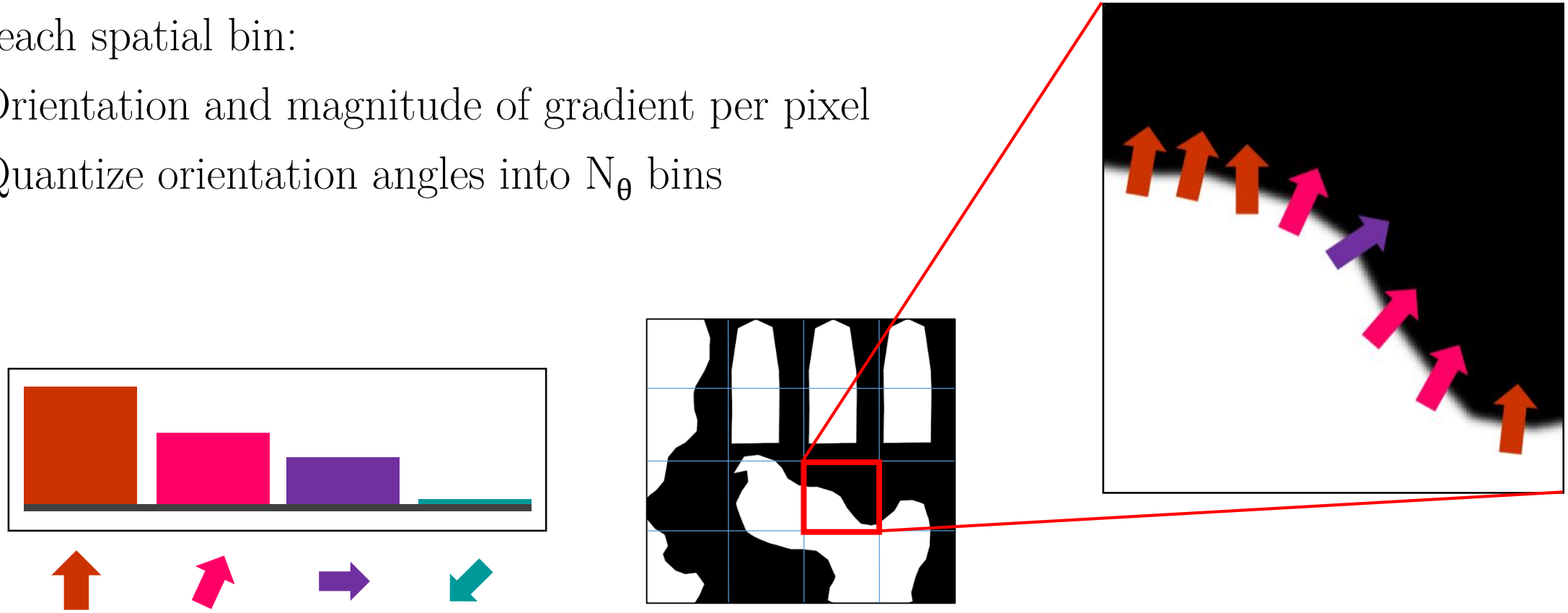
- Orientation and magnitude of gradient per pixel
- Quantize orientation angles into N_θ bins



SIFT (continued)

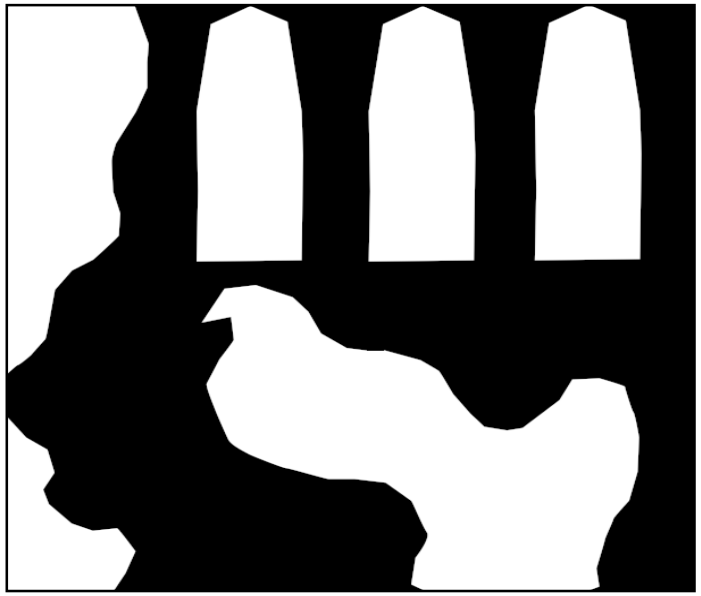
In each spatial bin:

- Orientation and magnitude of gradient per pixel
- Quantize orientation angles into N_θ bins



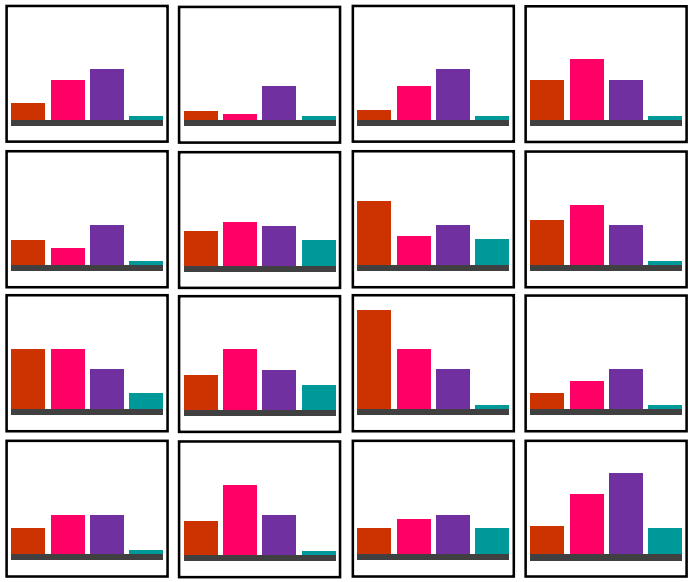
SIFT (continued)

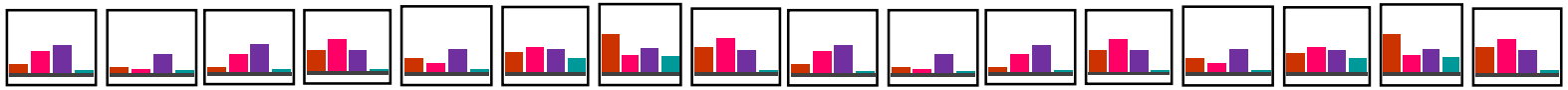
Descriptor: concatenation of histograms



SIFT (continued)

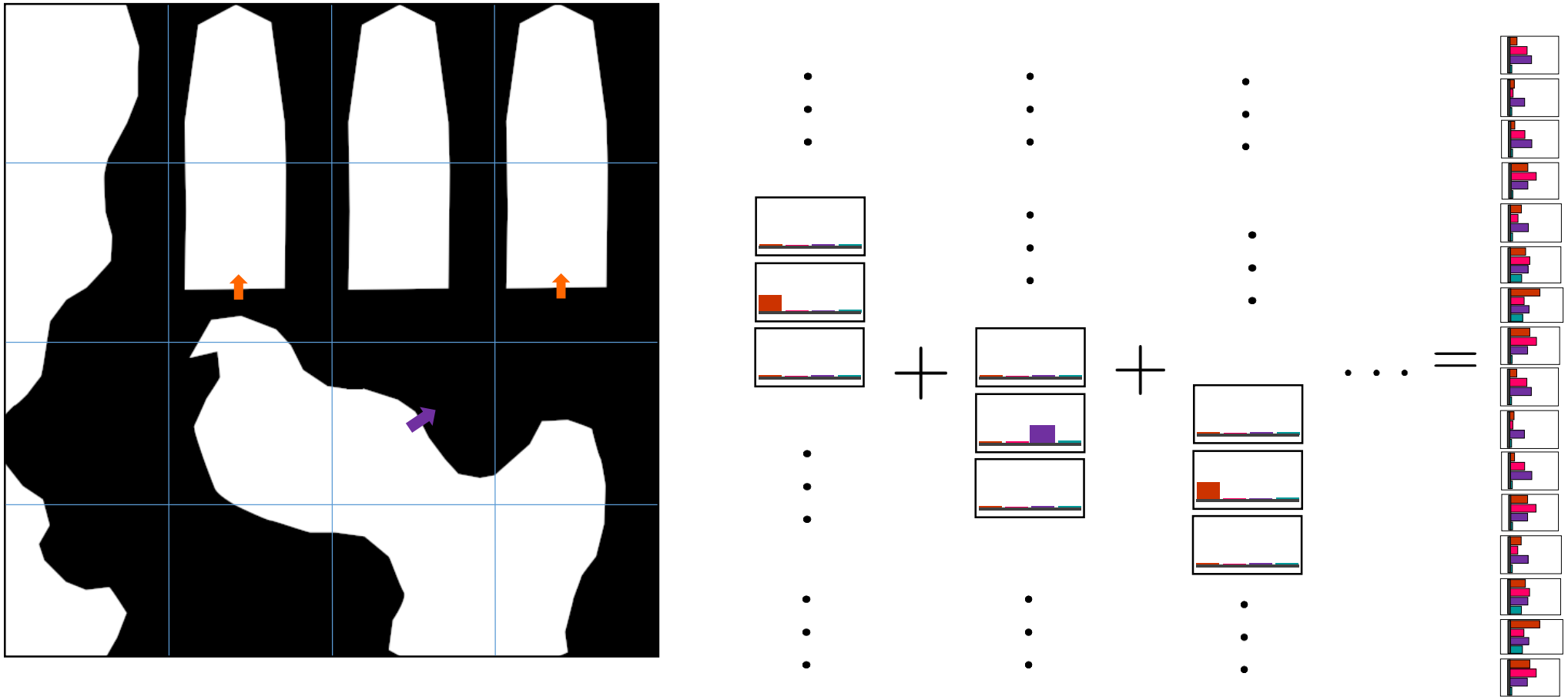
Descriptor: concatenation of histograms



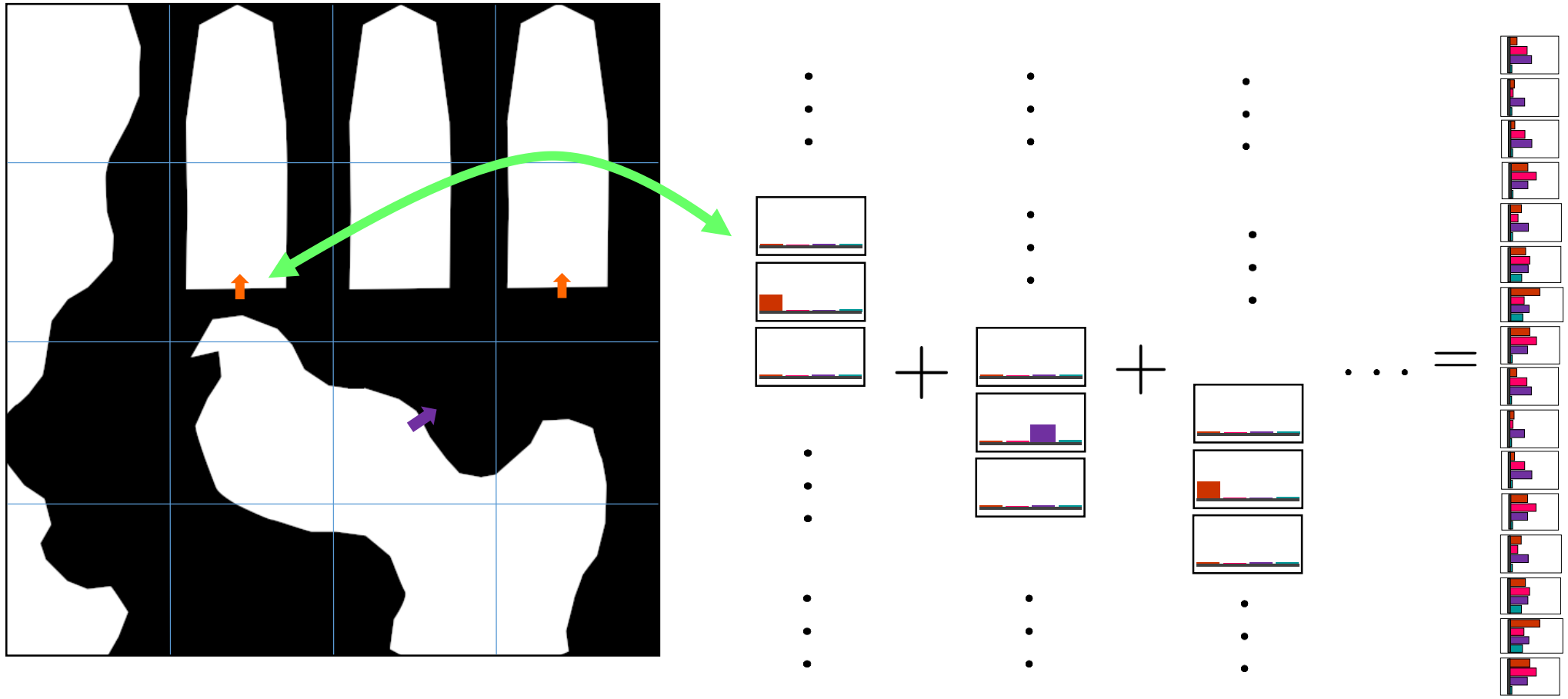


$4 \times 4 \times 8 = 128$ dimensional descriptor

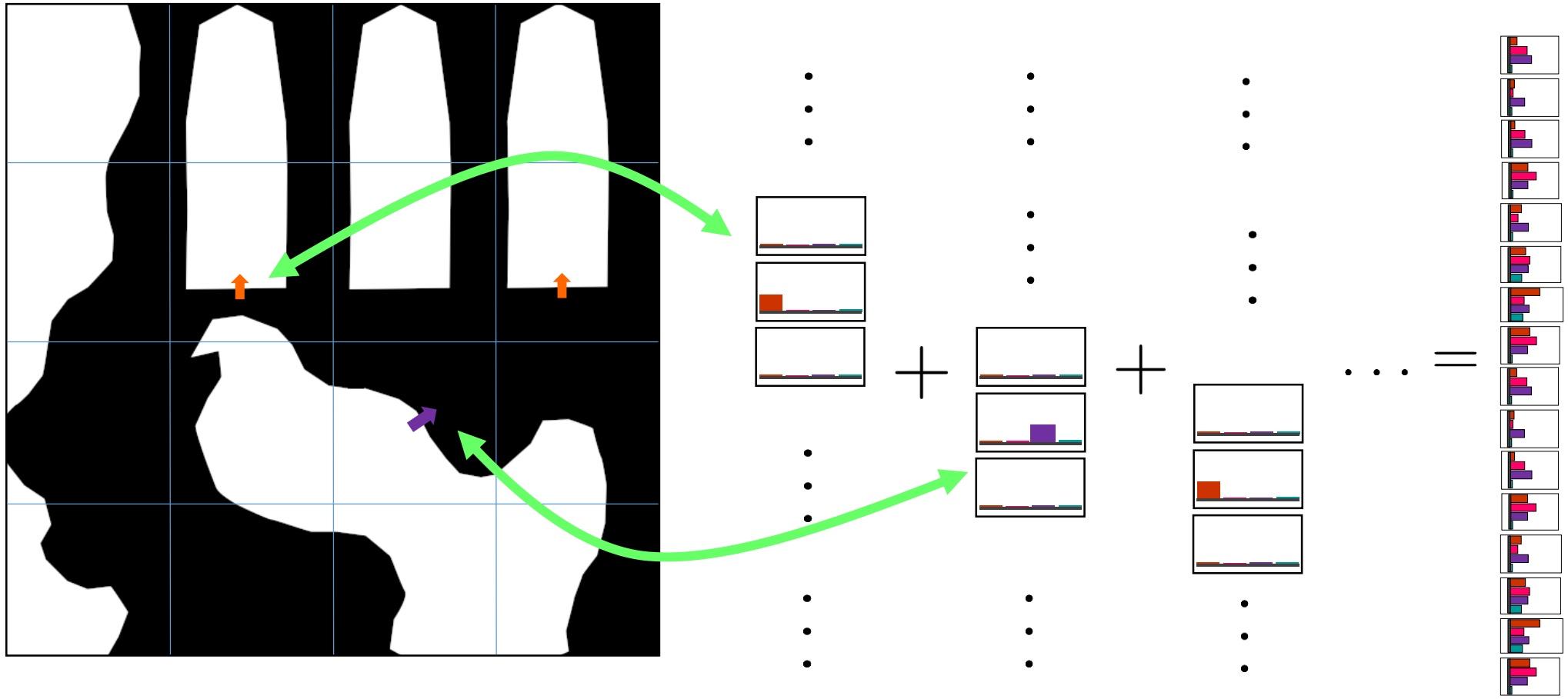
SIFT – contribution of a single pixel



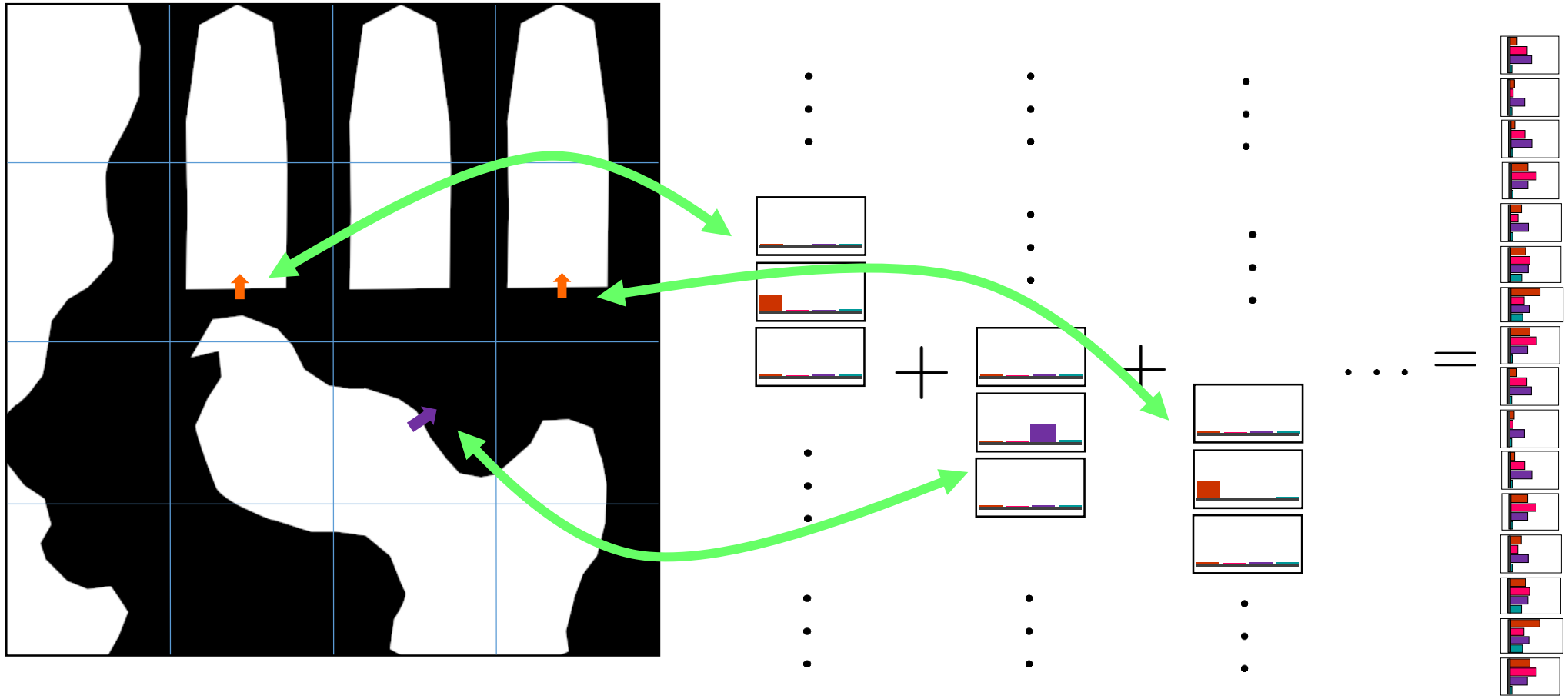
SIFT – contribution of a single pixel



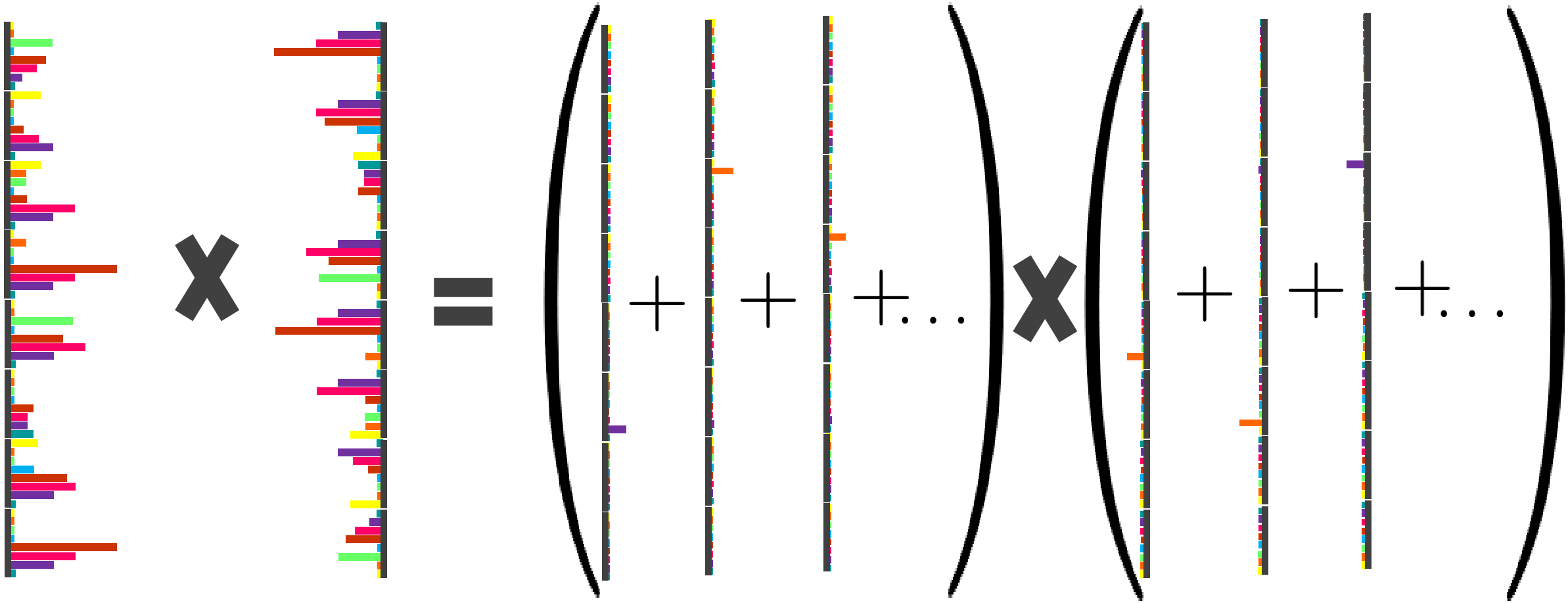
SIFT – contribution of a single pixel



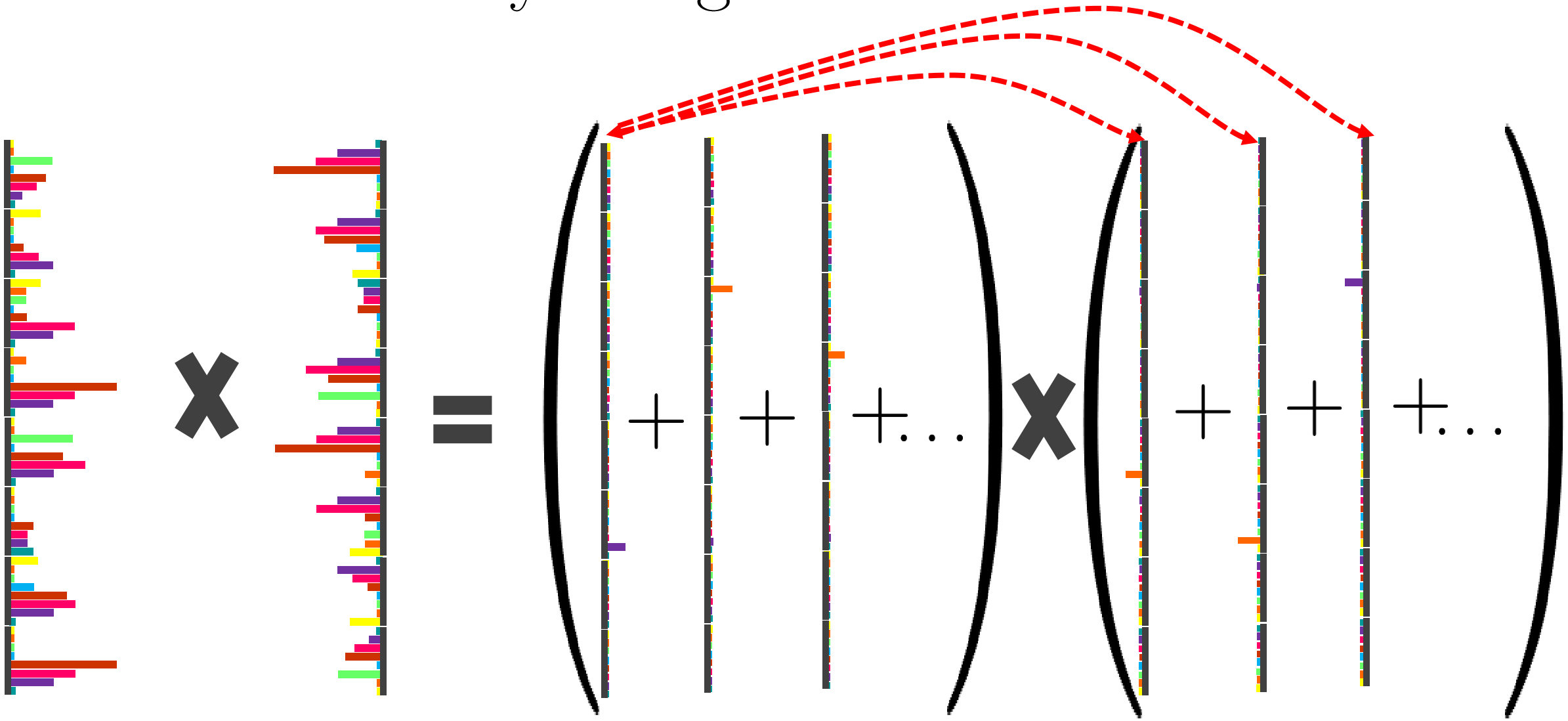
SIFT – contribution of a single pixel



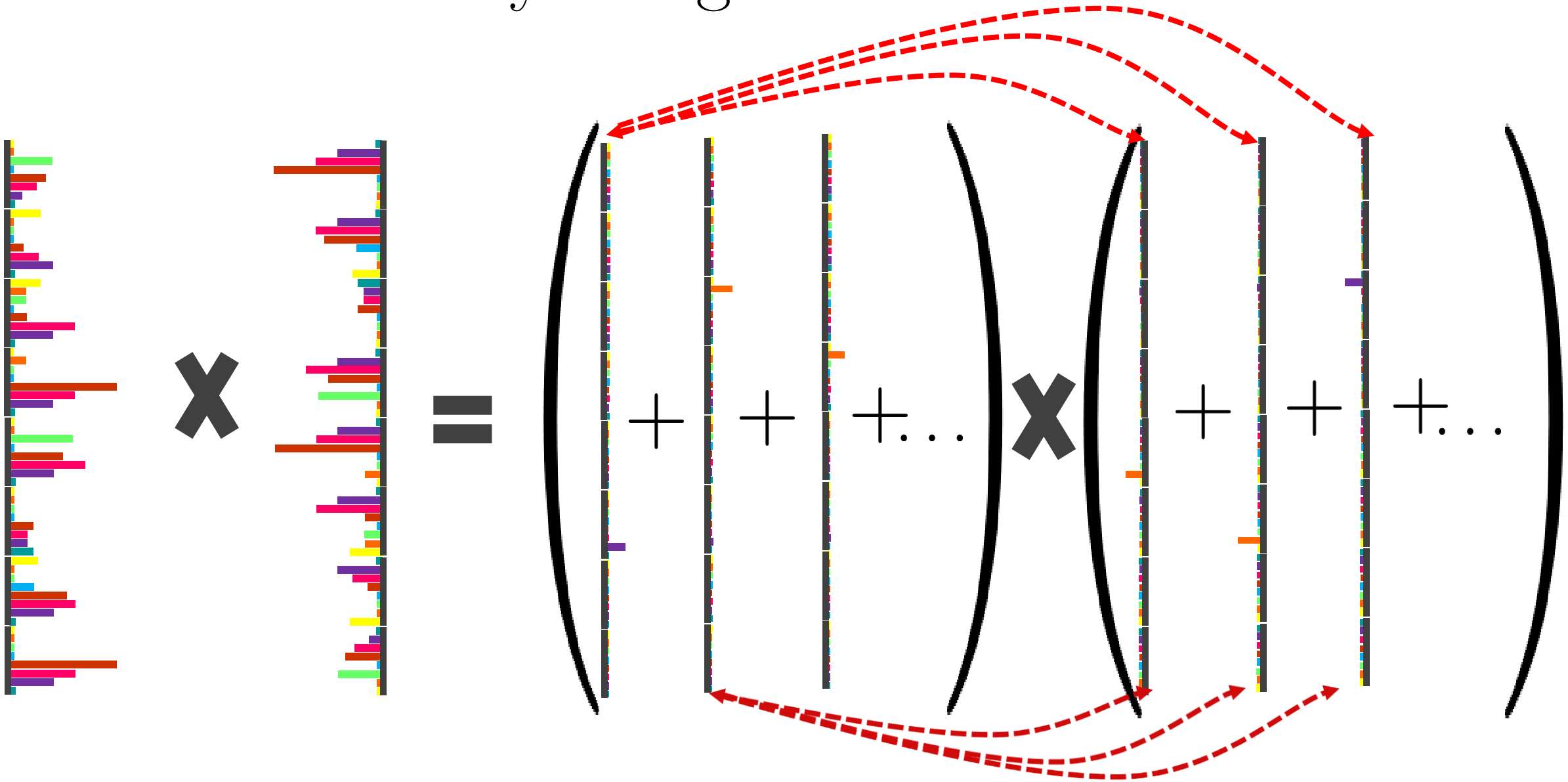
Patch similarity using SIFT



Patch similarity using SIFT



Patch similarity using SIFT



Patch similarity using SIFT

$$< \psi(\mathcal{P}), \psi(\mathcal{Q}) > \;=\; < \sum \psi(p), \sum \psi(q) > \;=\; \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{Q}} \psi(p)^\top \psi(q)$$

$$= \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{Q}} \boxed{k(p,q)}$$

SIFT

SIFT

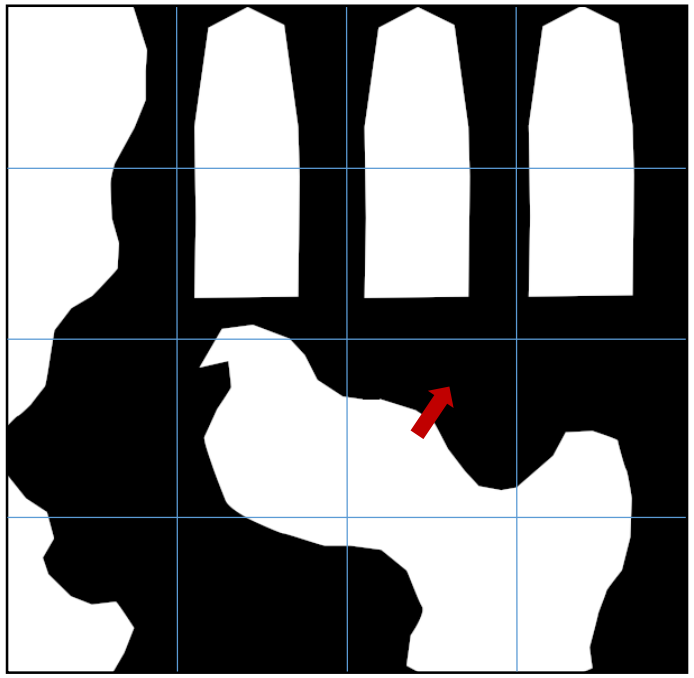
patch $\mathcal{P}$

patch $\mathcal{Q}$

SIFT similarity between two patches $\mathcal{P}$ and $\mathcal{Q}$

Patch similarity using SIFT

Compares all pixel pairs based on attributes x, y, θ

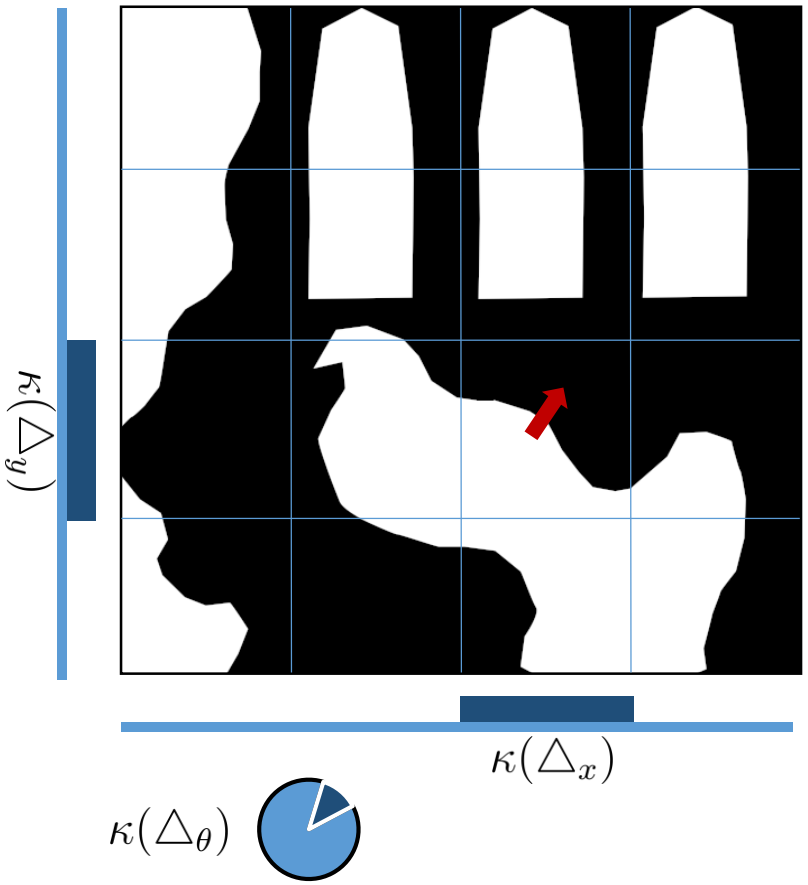


Patch similarity using SIFT

Compares all pixel pairs based on attributes x, y, θ

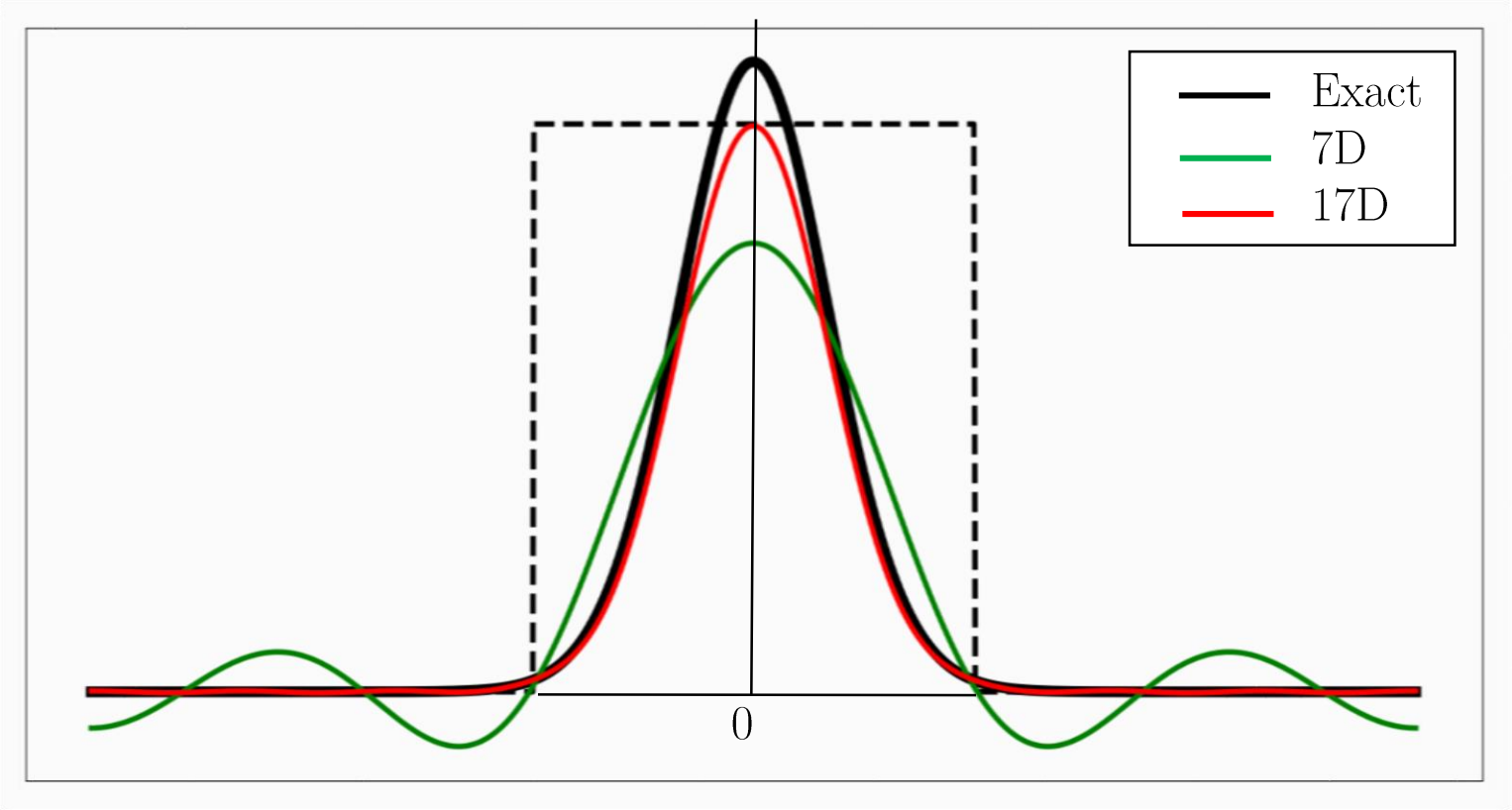
$$\boxed{\kappa(\triangle_x) \cdot \kappa(\triangle_y) \cdot \kappa(\triangle_\theta)}$$

$$\kappa(\triangle_x) = \begin{cases} 1 & \text{, same bin} \\ 0 & \text{, otherwise} \end{cases}$$



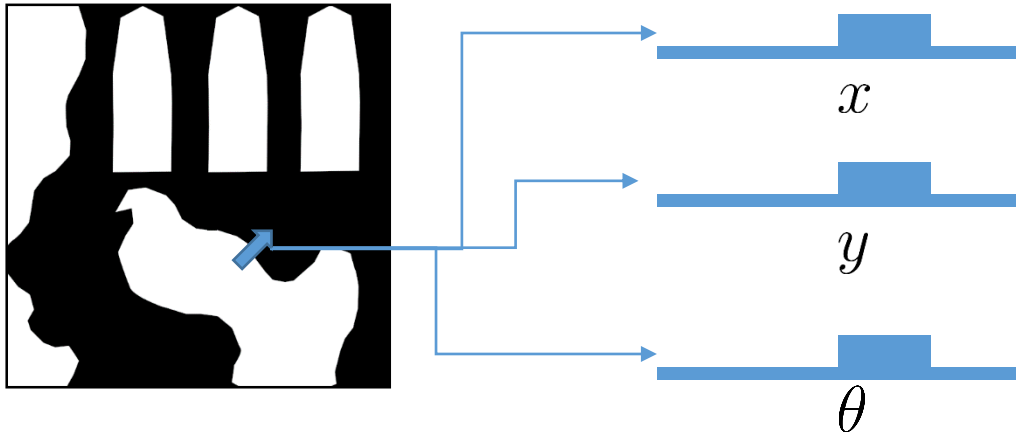
Extension to continuous similarity

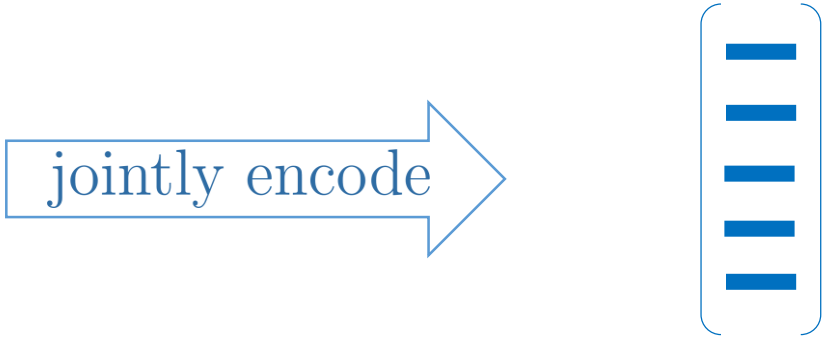
- Map 1D attribute to vector embedding
- Scalar product approximates von Mises kernel
- Dimensionality of embedding controls the approximation



[Bursuc et al. ICMR '15]

Pixel embedding

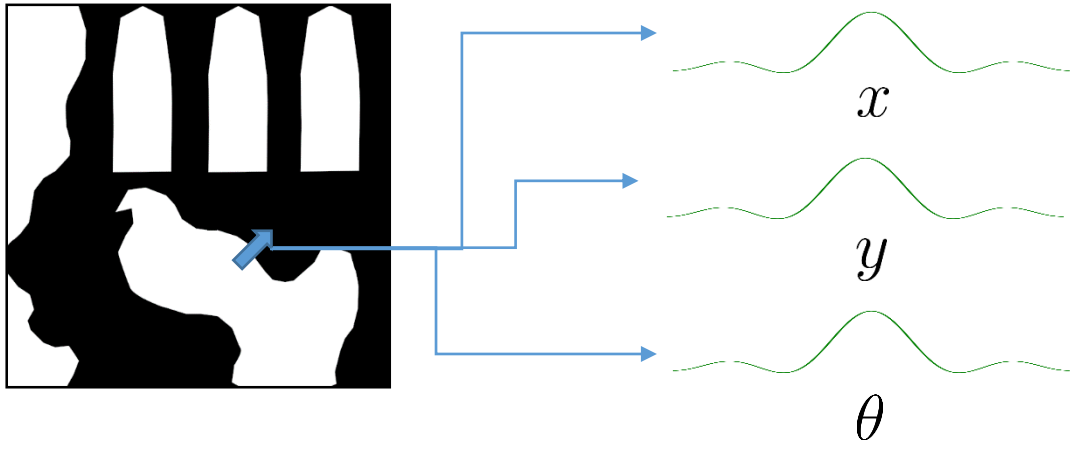


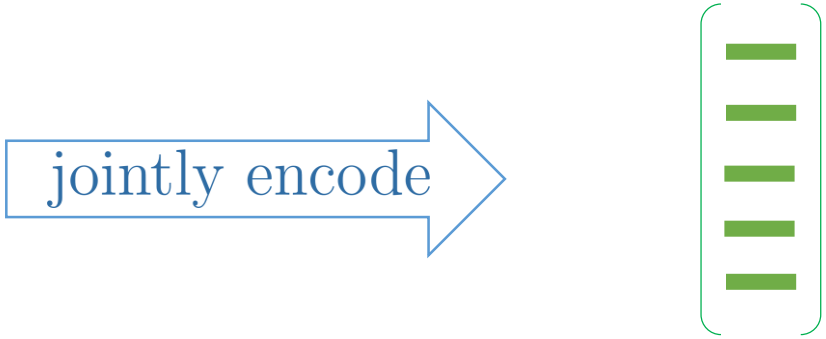


$$\psi(x, y, \theta)$$

SIFT

Pixel embedding





$$\psi(x, y, \theta)$$

Kernel descriptor

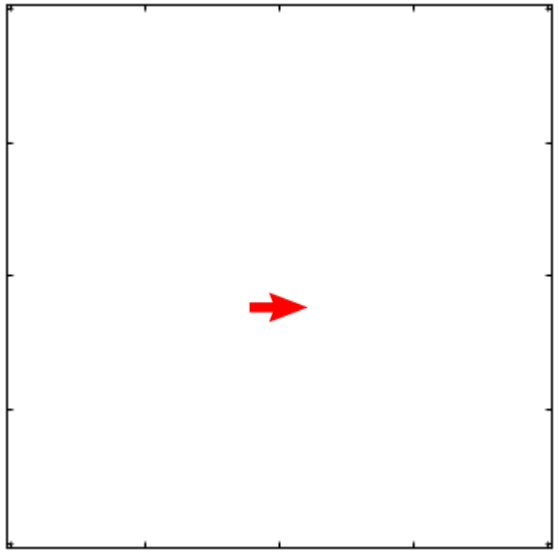
Visualizing patch similarity

- Pixel to pixel similarity

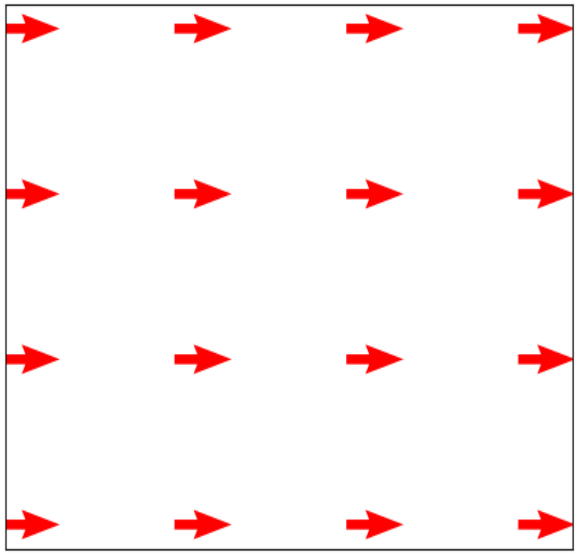
$$k(p,q) = < \psi(p), \psi(q) >$$
$$\approx k_x \cdot k_y \cdot k_\theta$$

- Patchmap: visualize contribution over the whole patch

Pixel p



Patch $\mathcal{Q}$



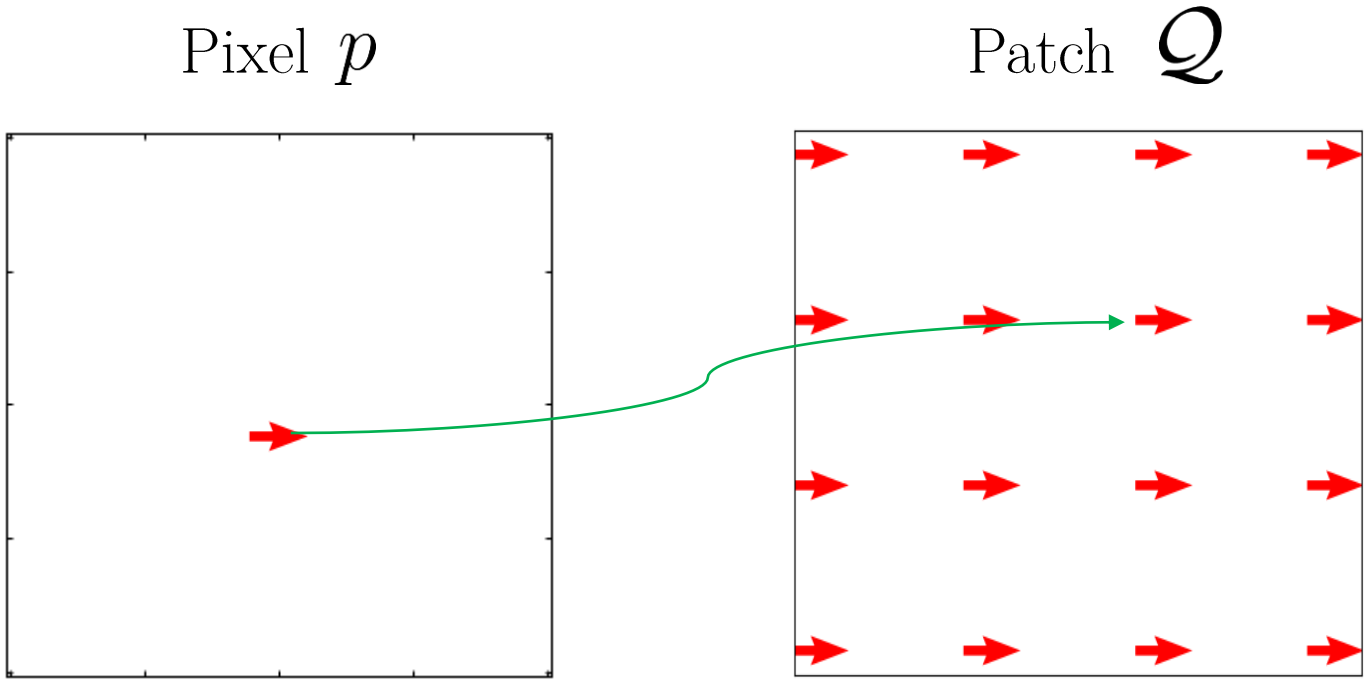
Gradient angle shown by

Visualizing patch similarity

- Pixel to pixel similarity

$$k(p,q) = < \psi(p), \psi(q) >$$
$$\approx k_x \cdot k_y \cdot k_\theta$$

- Patchmap: visualize contribution over the whole patch



Gradient angle shown by

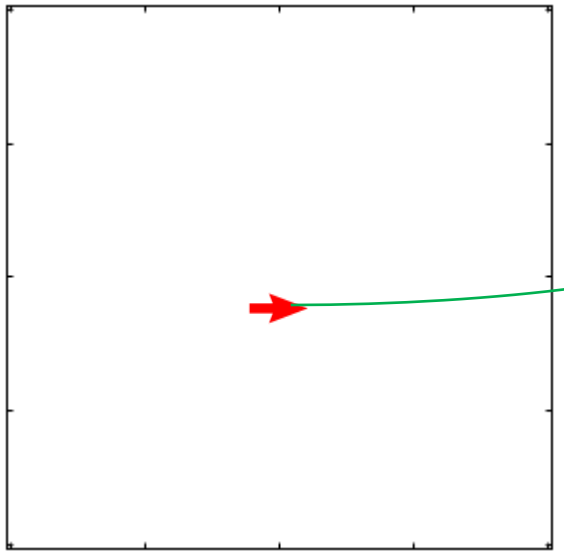
Visualizing patch similarity

- Pixel to pixel similarity

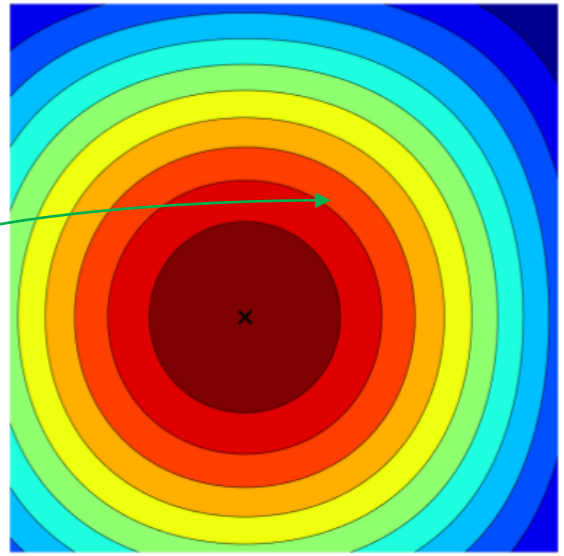
$$k(p,q) = < \psi(p), \psi(q) >$$
$$\approx k_x \cdot k_y \cdot k_\theta$$

- Patchmap: visualize contribution over the whole patch

Pixel p



Patch $\mathcal{Q}$



10 iso-contours shown

Maximum similarity in red

Minimum similarity in blue

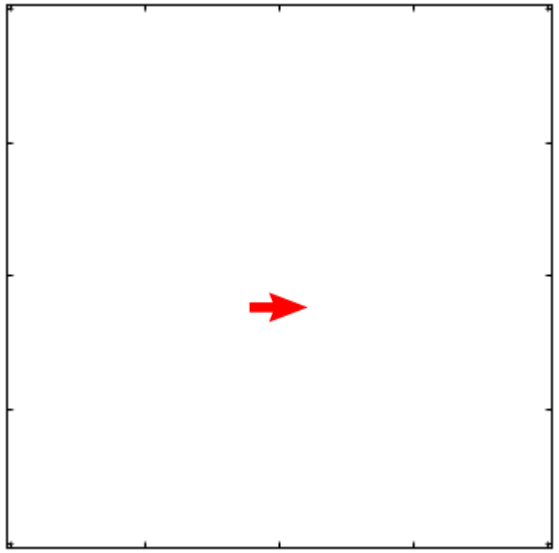
Visualizing patch similarity

- Pixel to pixel similarity

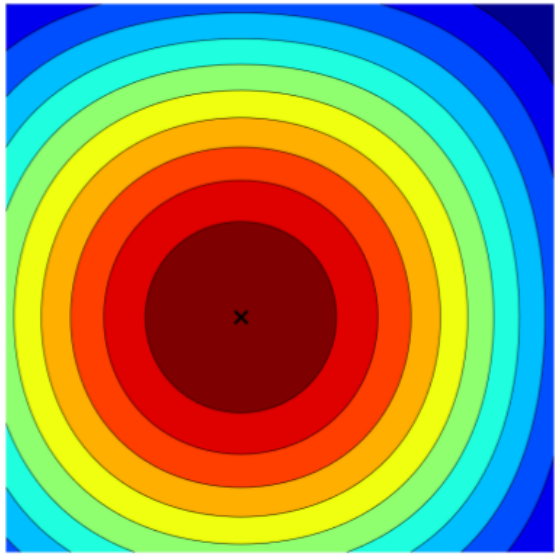
$$k(p,q) = < \psi(p), \psi(q) >$$
$$\approx k_x \cdot k_y \cdot k_\theta$$

- Patchmap: visualize contribution over the whole patch

Pixel p



Patch $\mathcal{Q}$



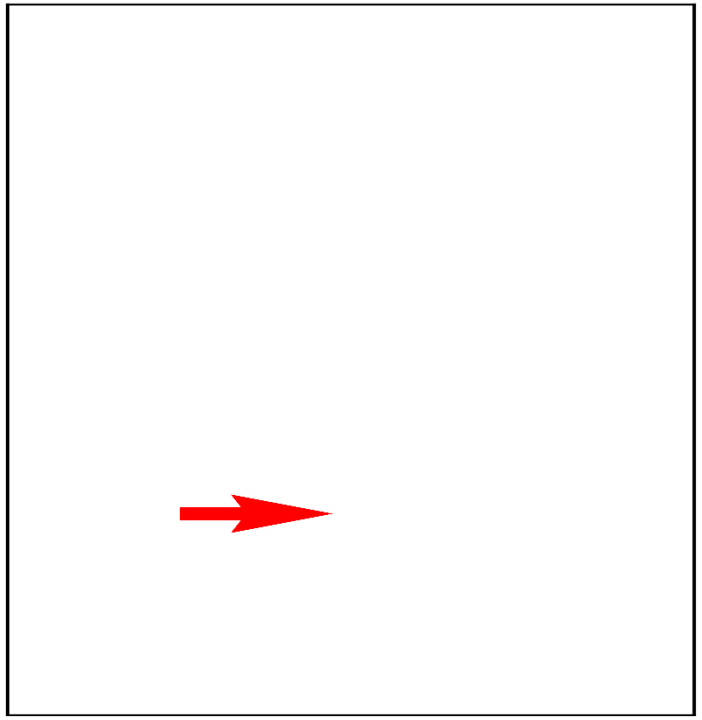
10 iso-contours shown

Maximum similarity in red

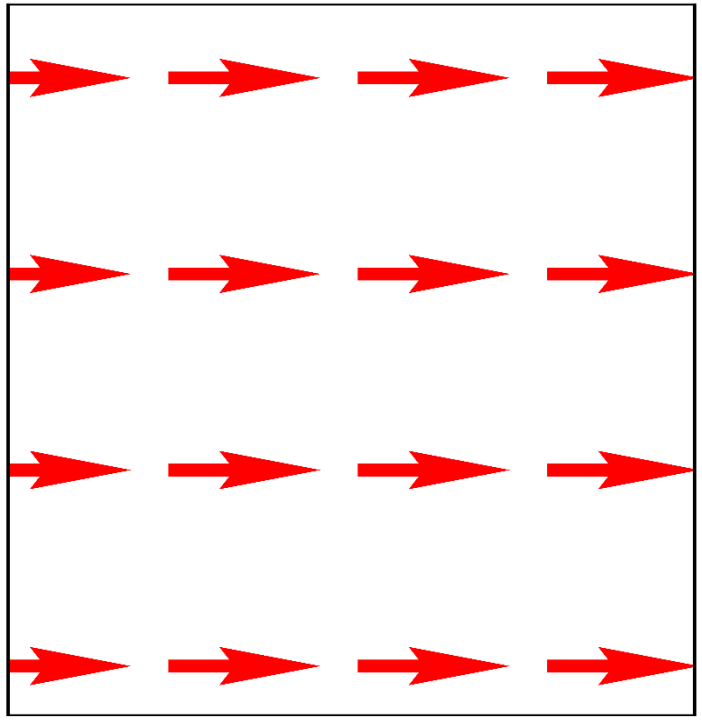
Minimum similarity in blue

Shift invariant similarity

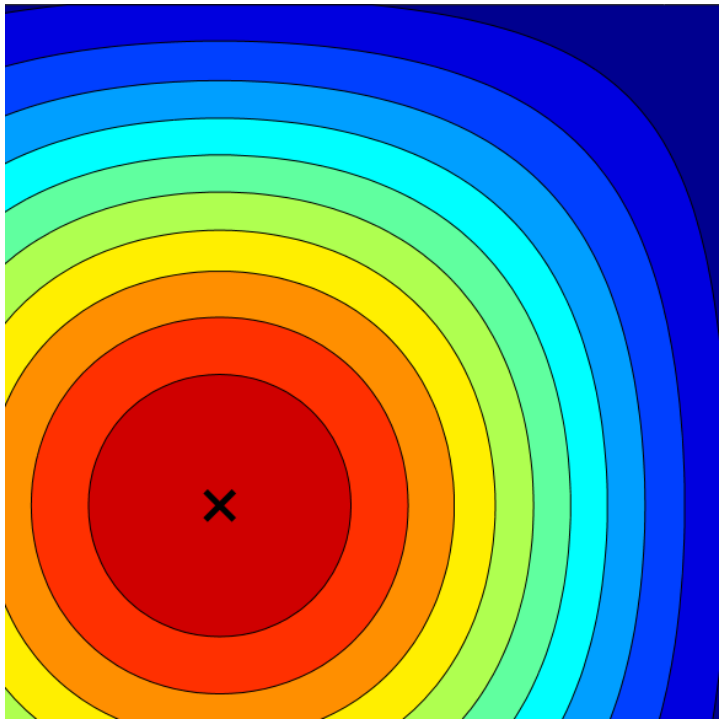
Stationary kernel: depends on the difference, not the absolute value



Pixel p

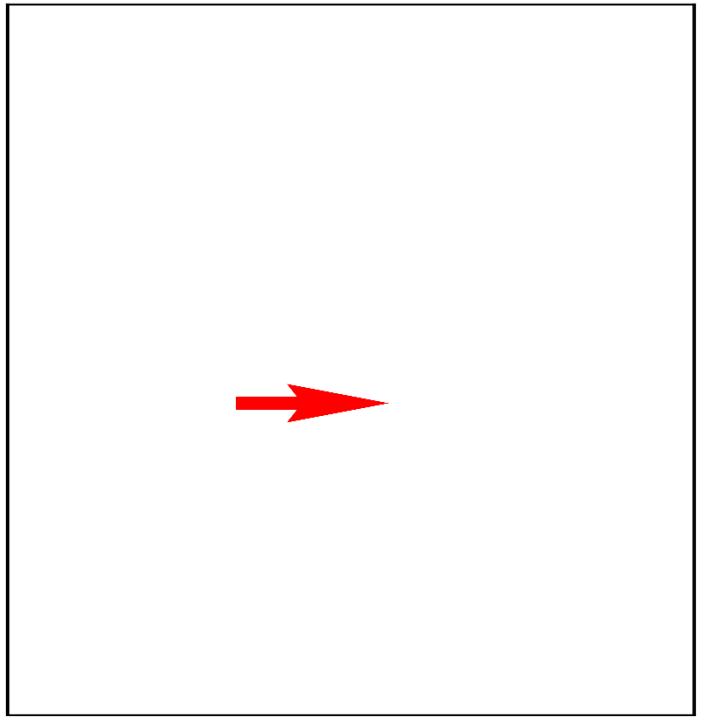


Patch $\mathcal{Q}$

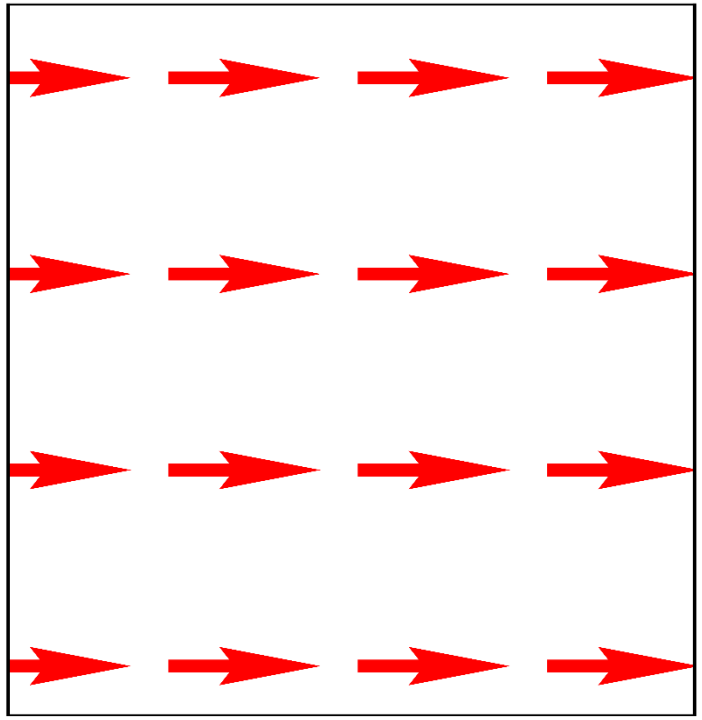


Shift invariant similarity

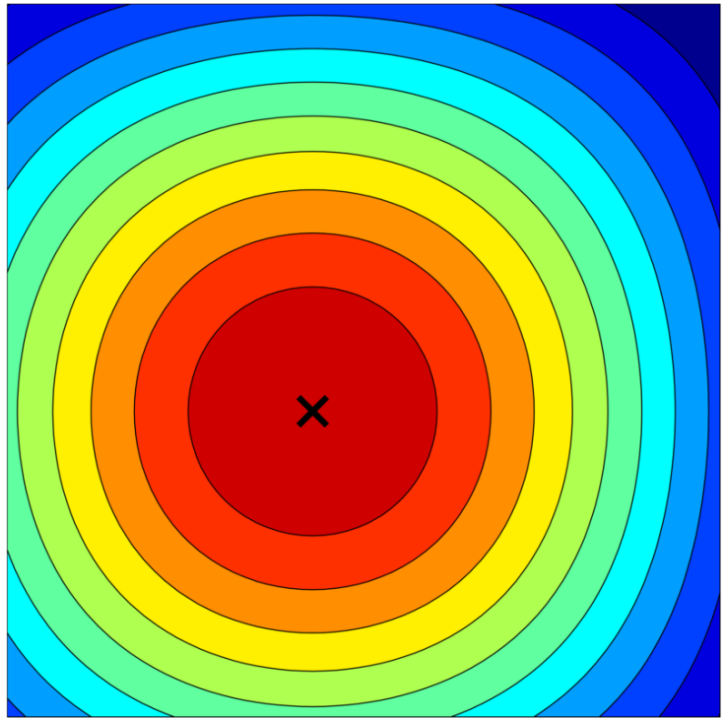
Stationary kernel: depends on the difference, not the absolute value



Pixel p

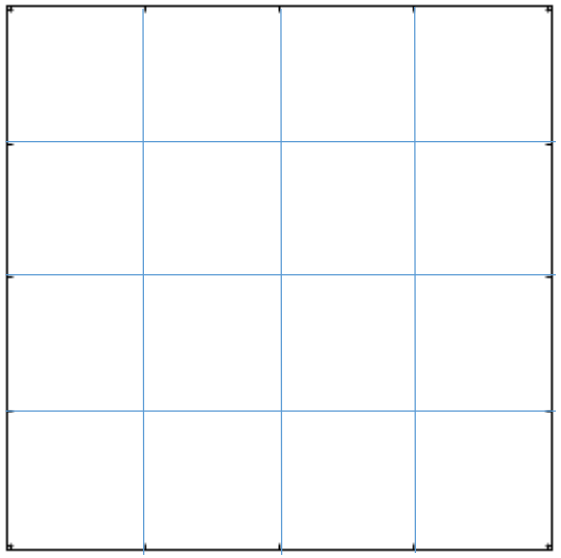


Patch $\mathcal{Q}$

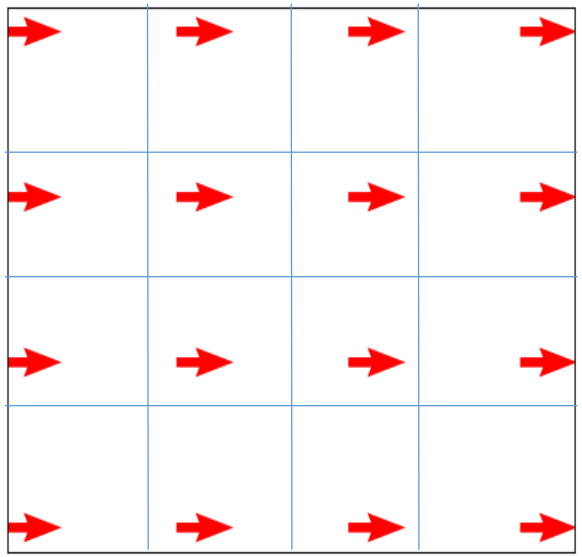


Patchmap for SIFT

Pixel p

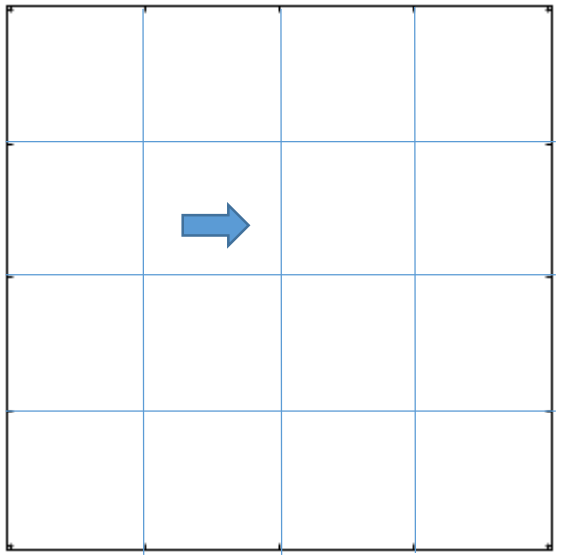


Patch $\mathcal{Q}$

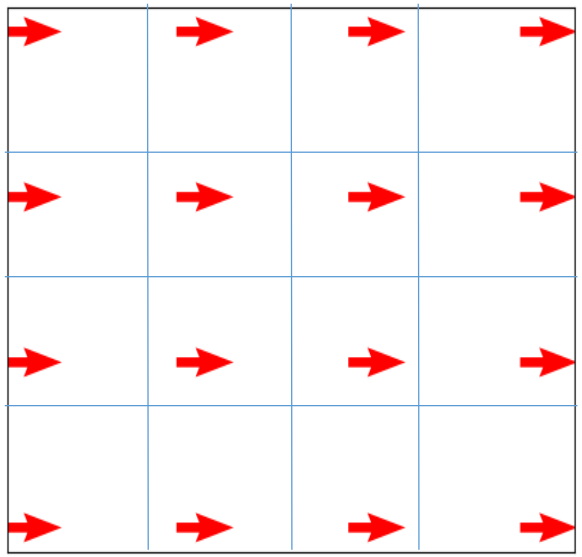


Patchmap for SIFT

Pixel p

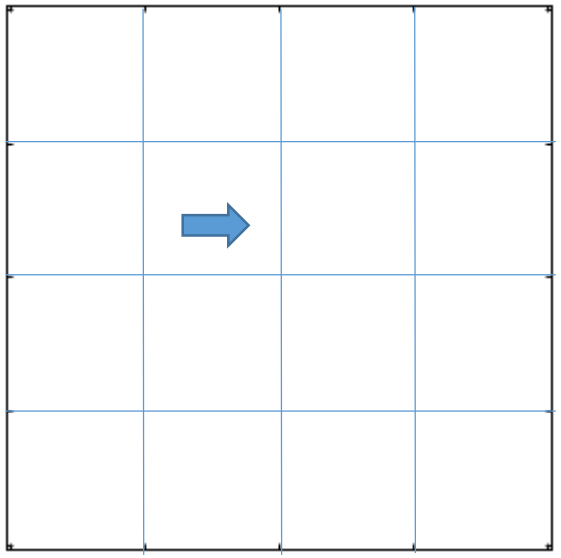


Patch $\mathcal{Q}$

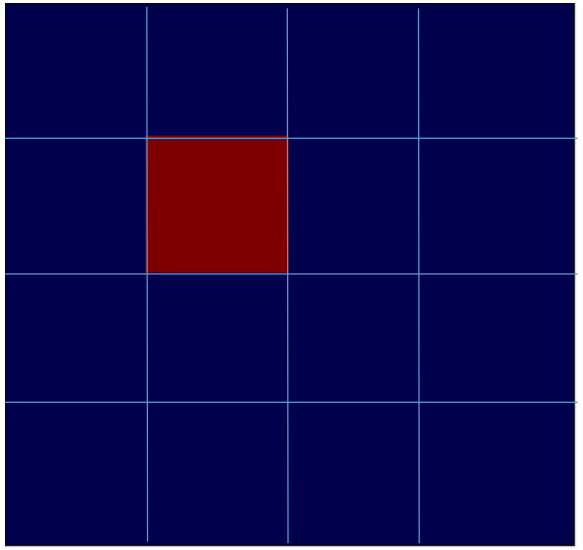


Patchmap for SIFT

Pixel p

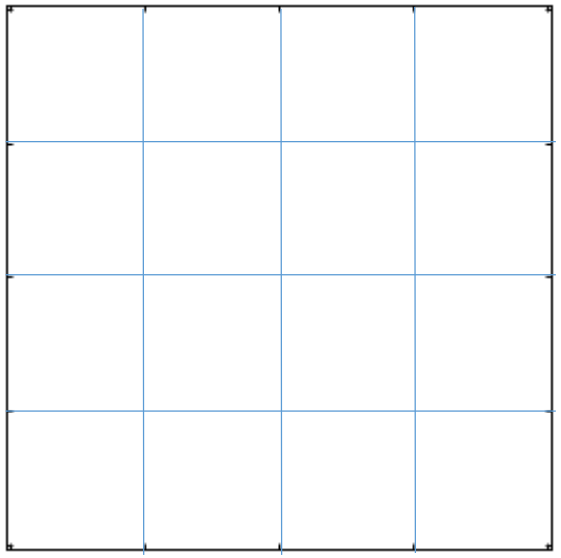


Patch $\mathcal{Q}$

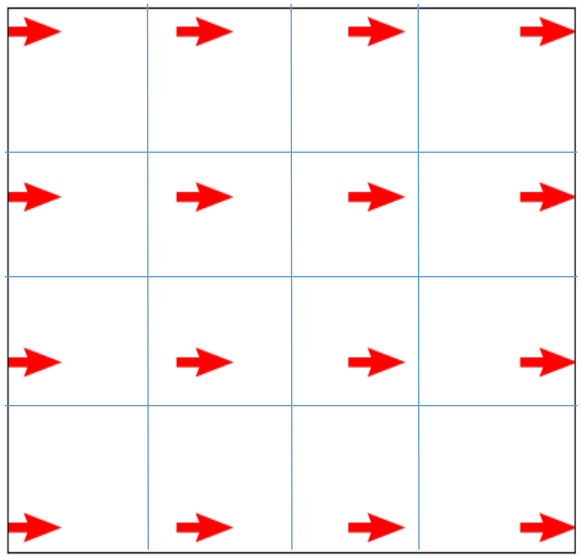


Patchmap for SIFT

Pixel p

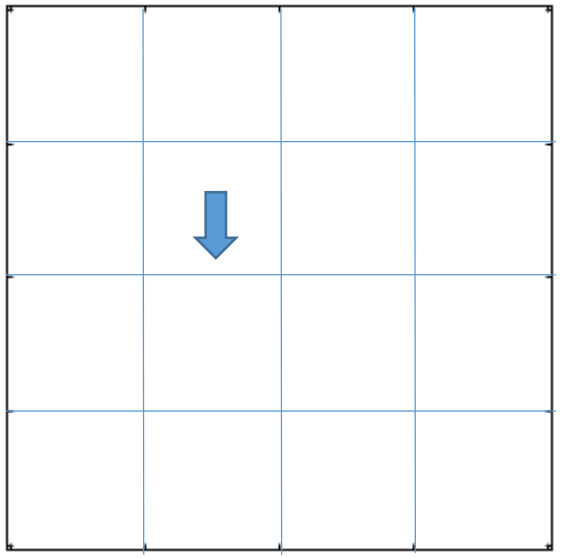


Patch $\mathcal{Q}$

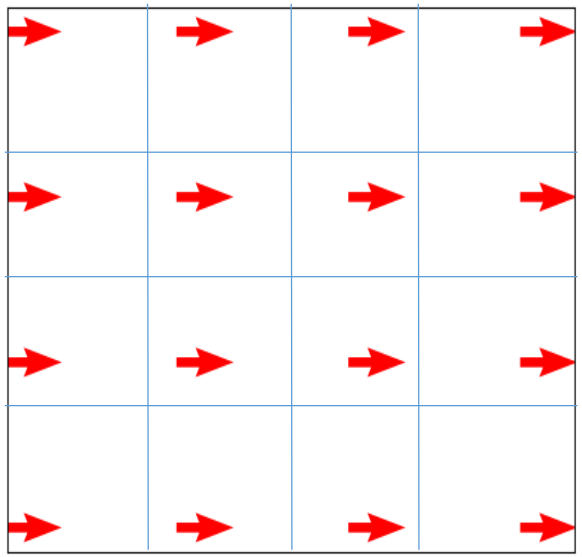


Patchmap for SIFT

Pixel p

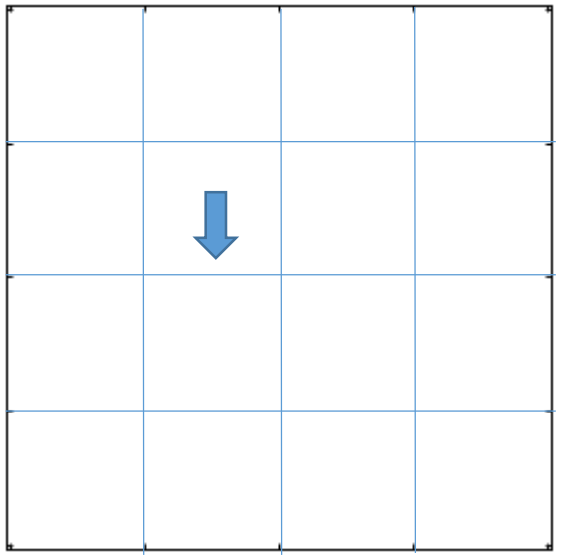


Patch $\mathcal{Q}$

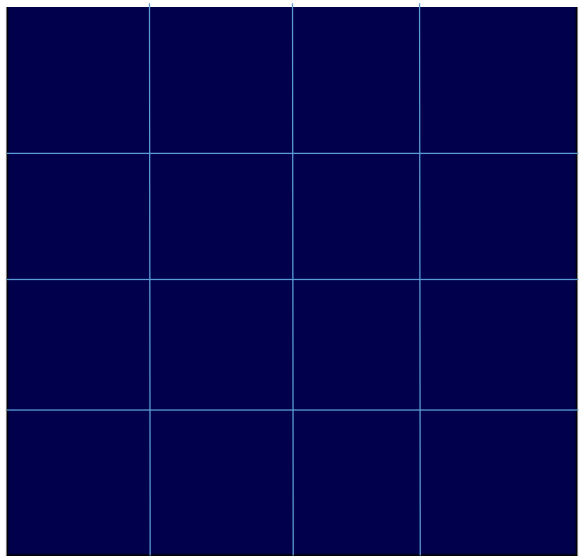


Patchmap for SIFT

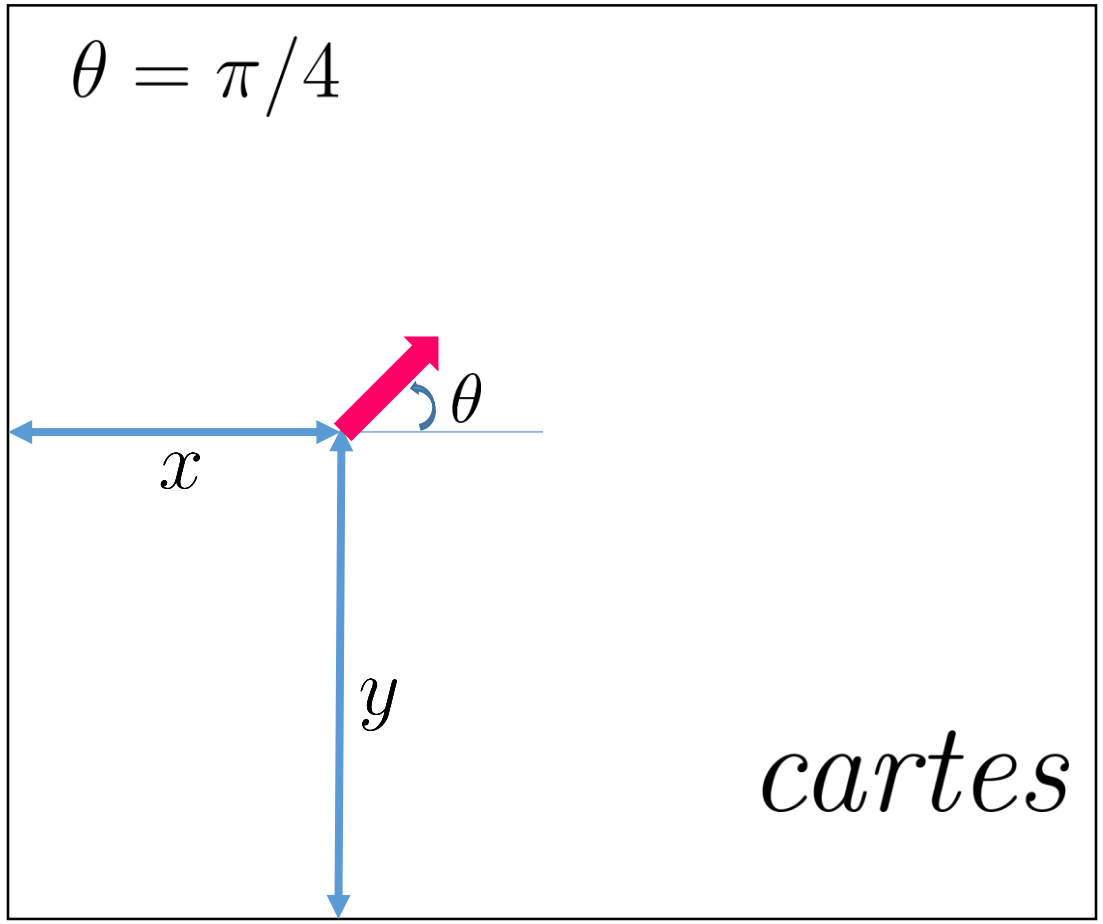
Pixel p

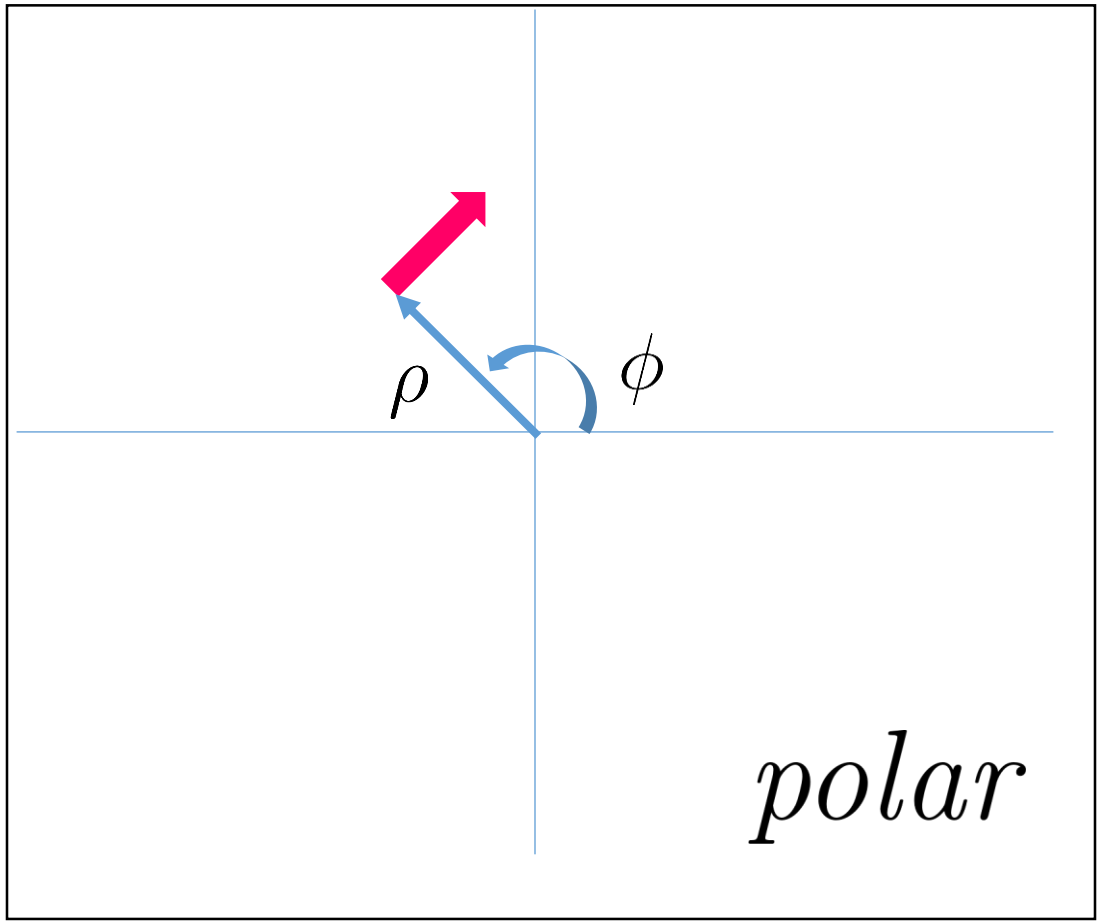


Patch $\mathcal{Q}$

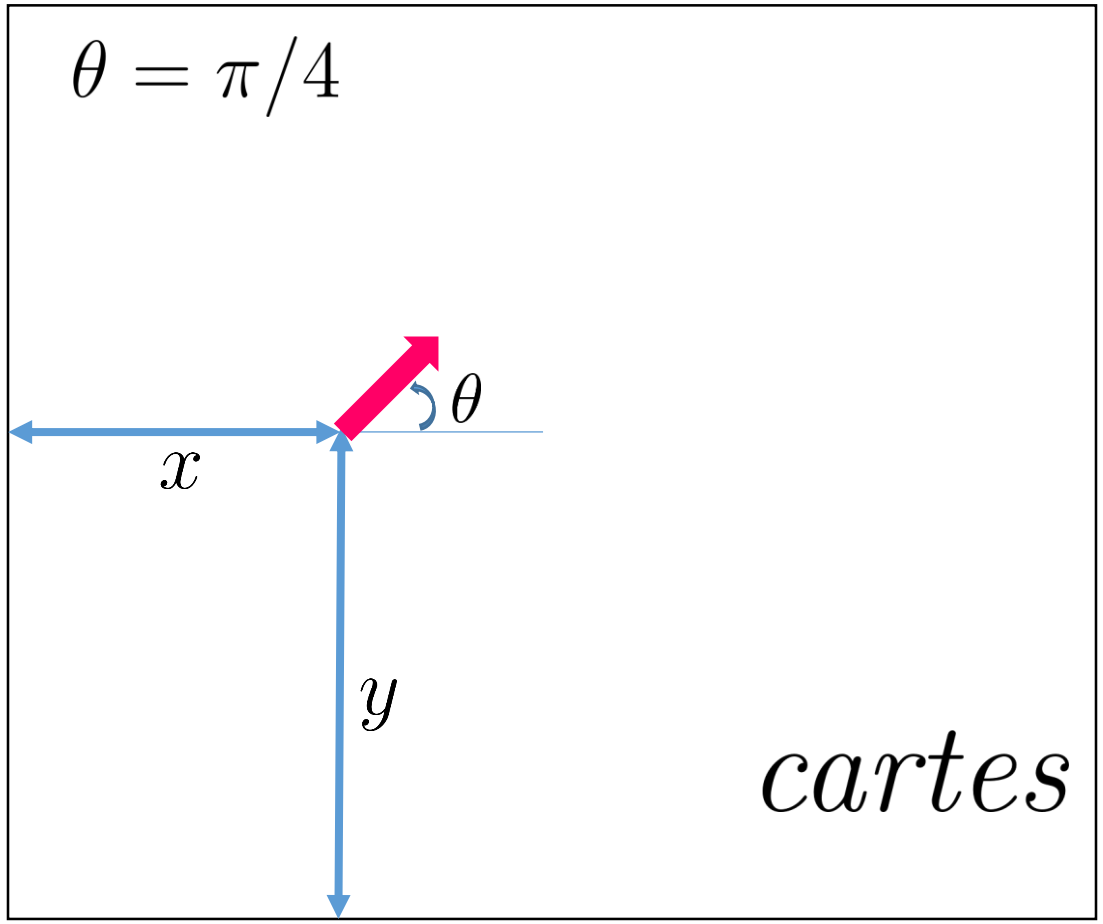


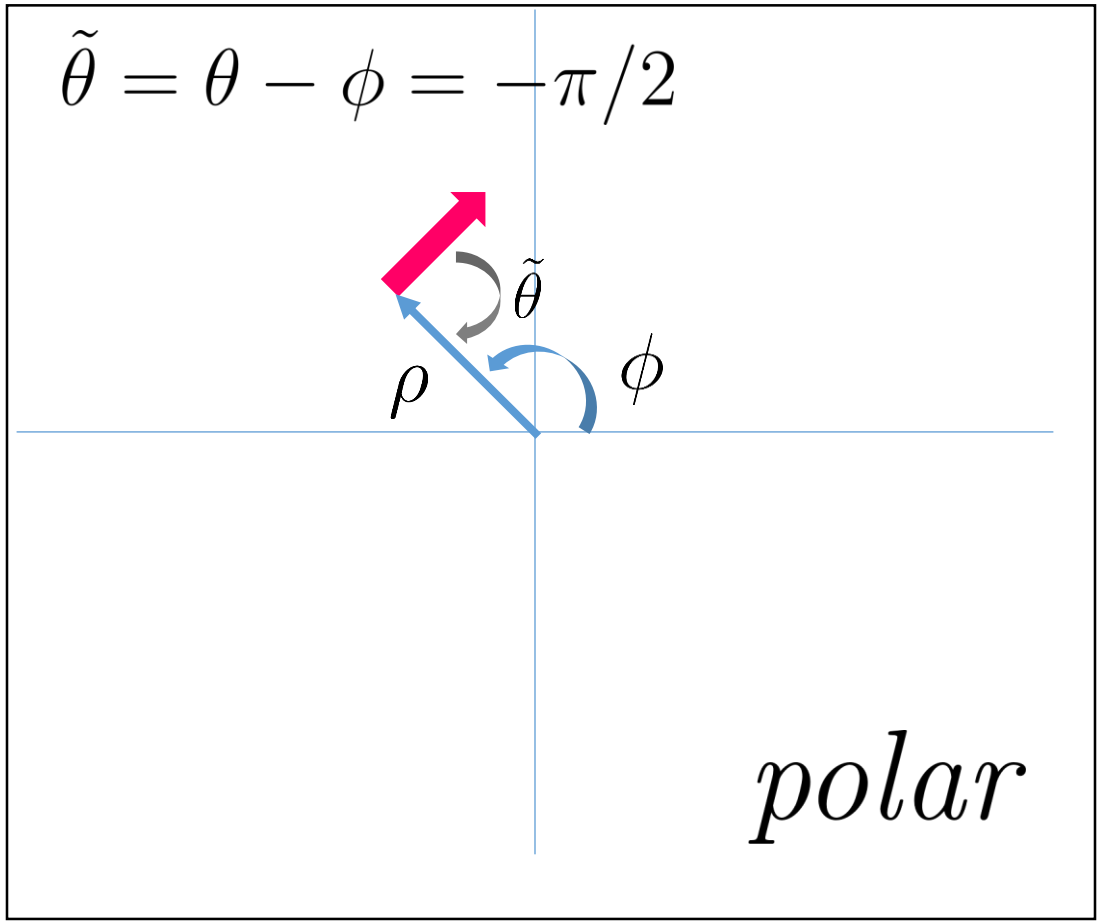
Patch parametrizations



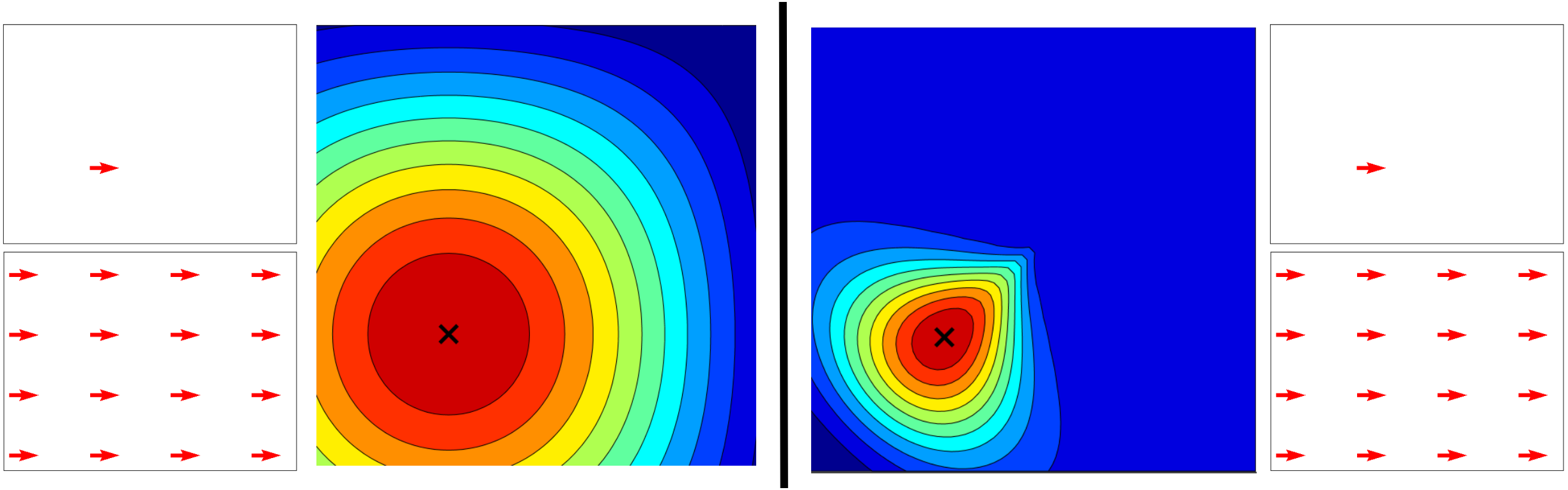


Patch parametrizations



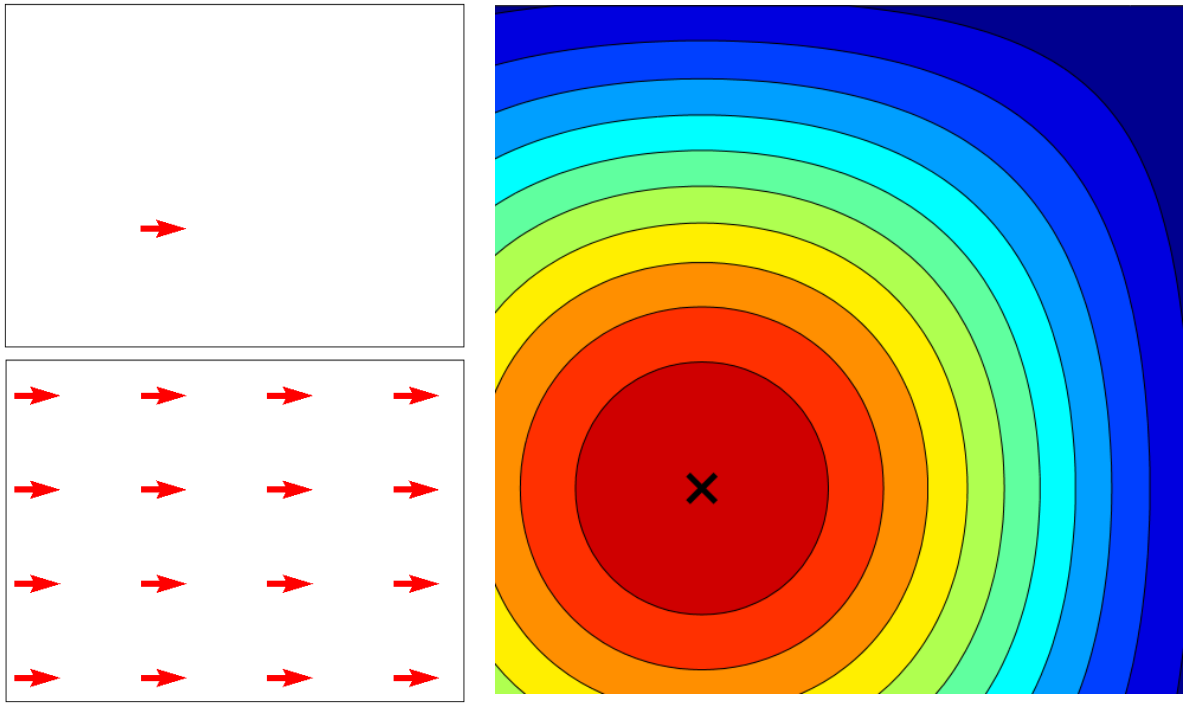


Patch parametrizations



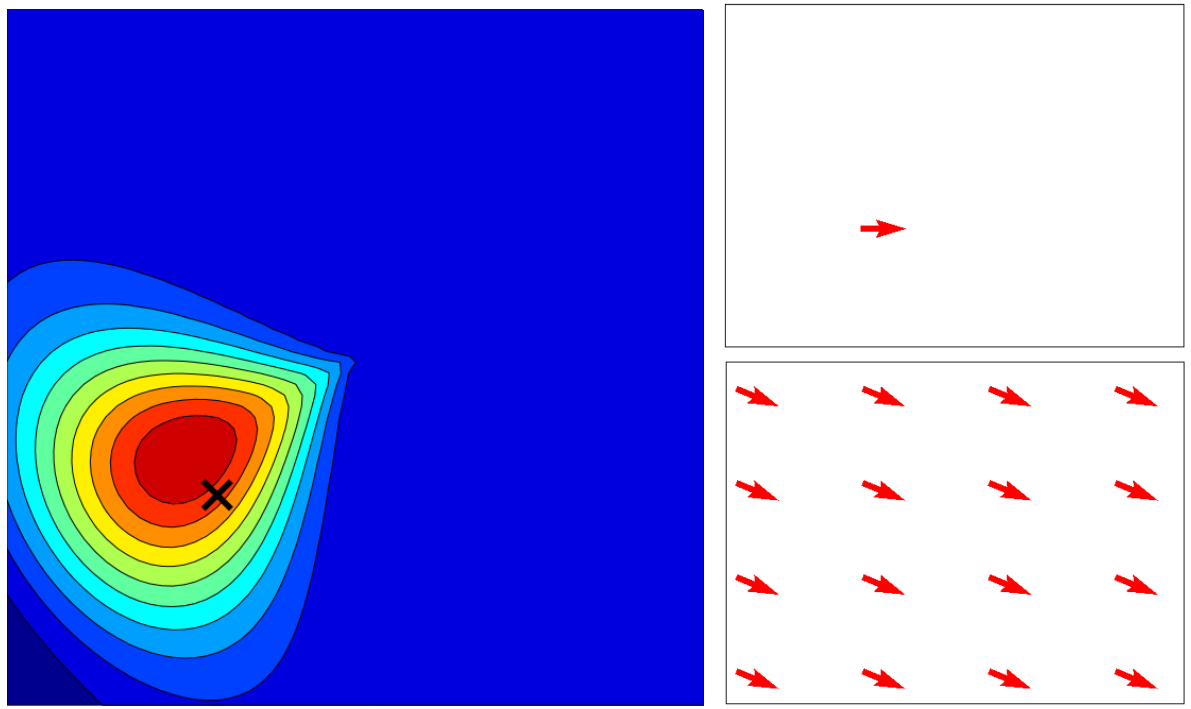
cartes *polar*

Patch parametrizations



cartes

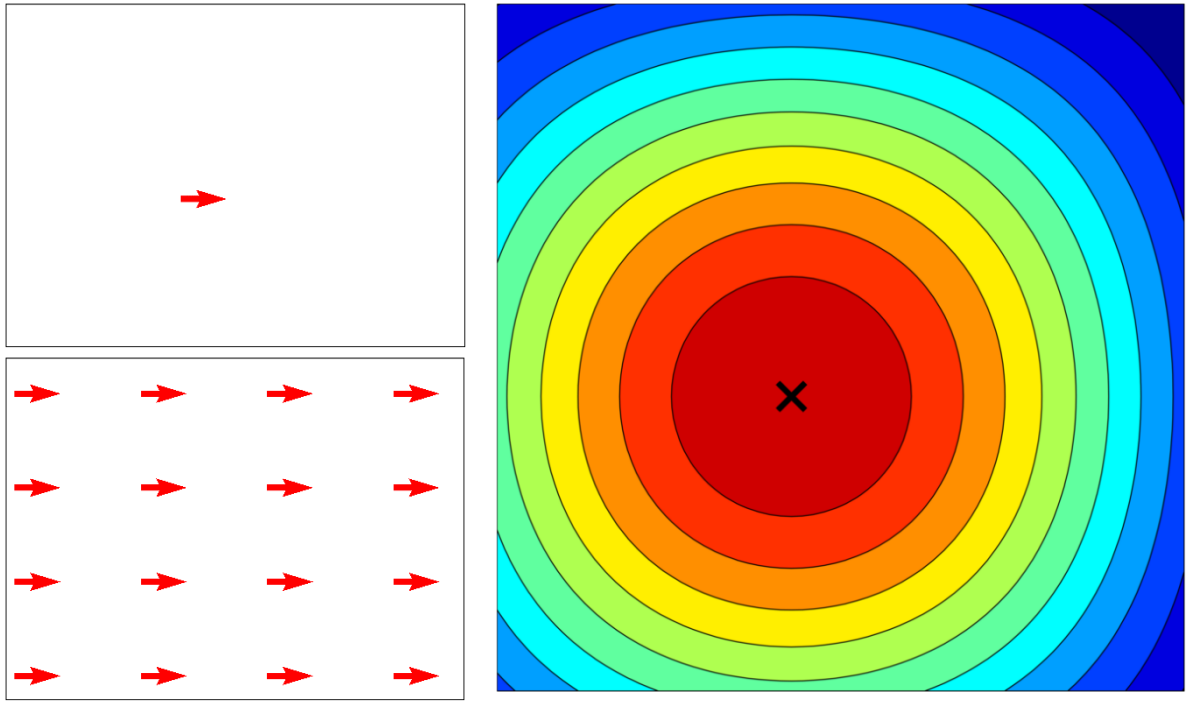
dominant orientation misalignment



polar

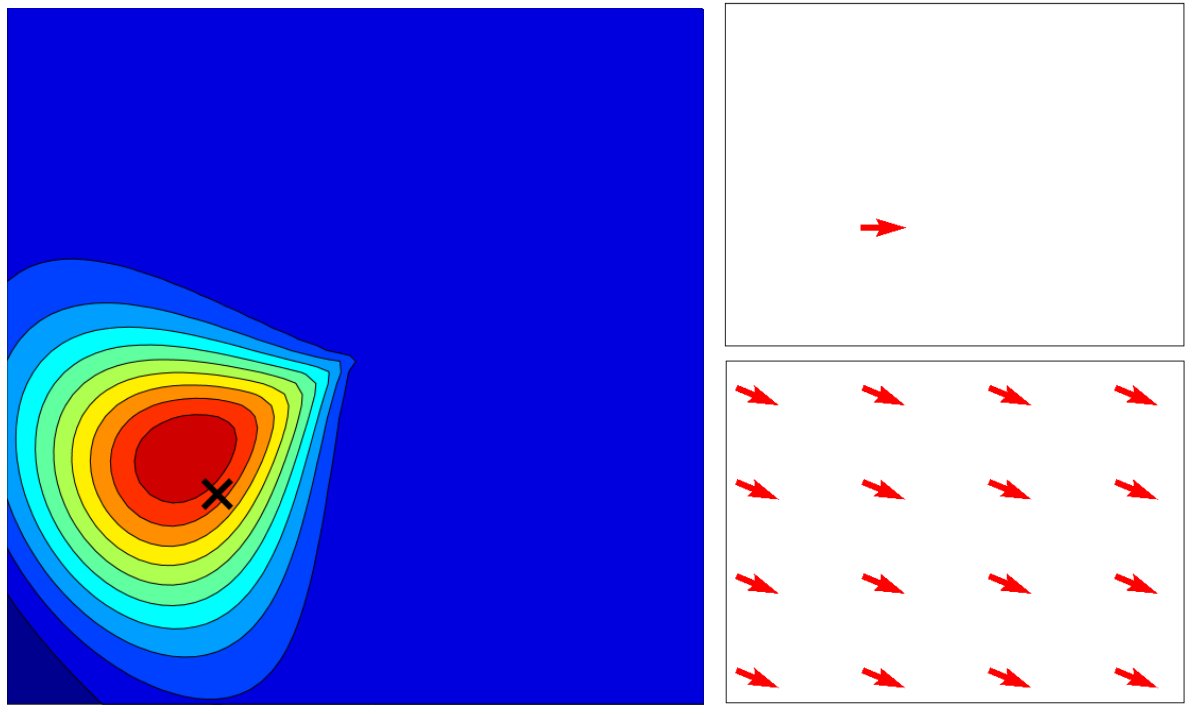
Patch parametrizations

keypoint center misalignment



cartes

dominant orientation misalignment



polar

Multiple parametrizations

Polar

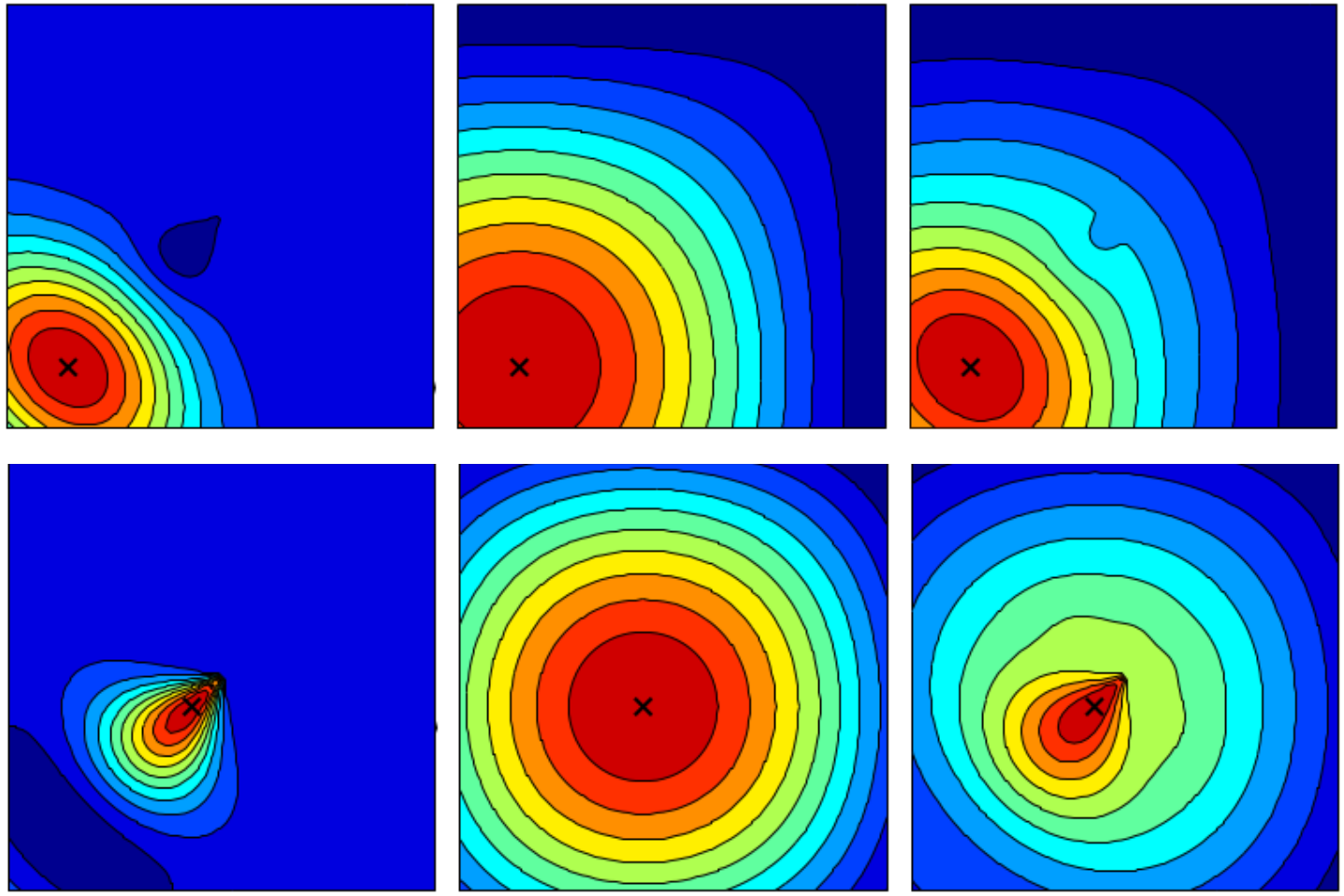
- Known to perform well [Bursuc ICMR '15]
- Tolerance to orientation mistakes
- Discontinuity around the center

Cartes

- Stable around the center
- Tolerance to location mistakes

We preserve both behaviors

- Simple descriptor concatenation

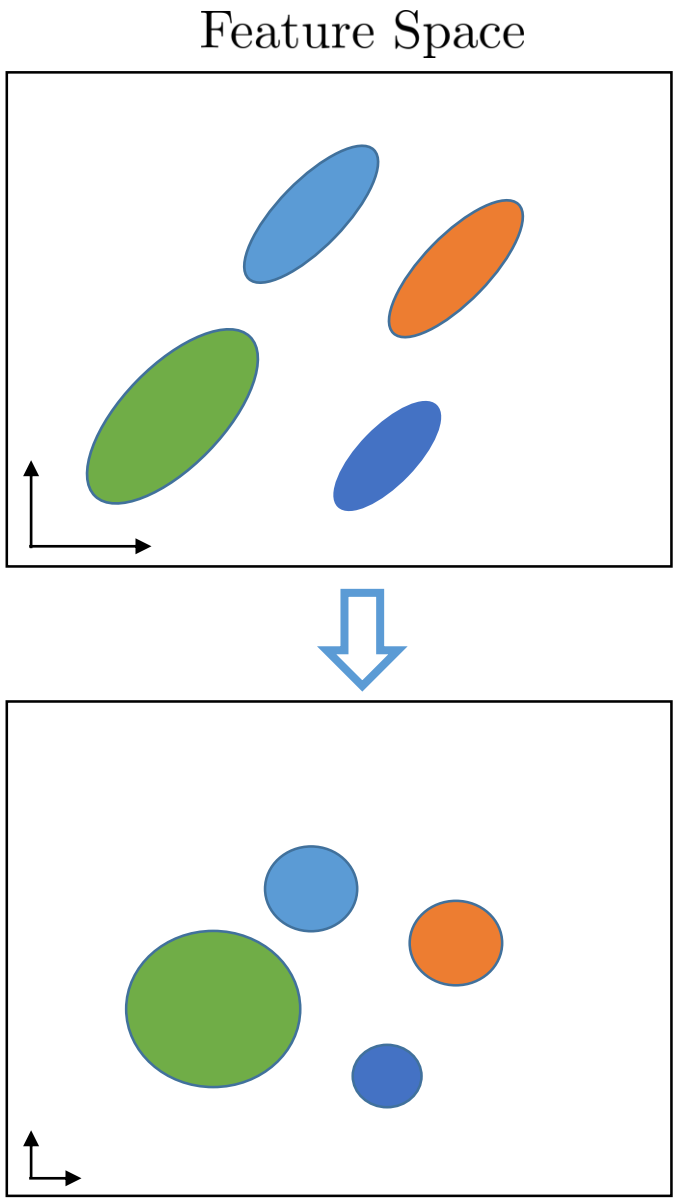


Polar Cartes Both

Injecting supervision

Linear discriminant projections

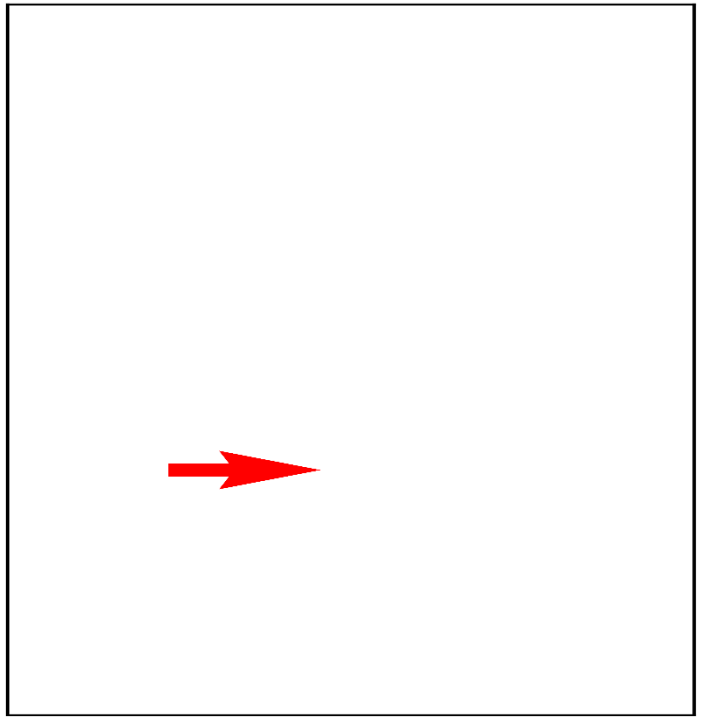
- Supervised descriptor whitening
- Very fast to learn!
- Works with few training examples
- Learns optimal mixing coefficient in case of concatenated vectors



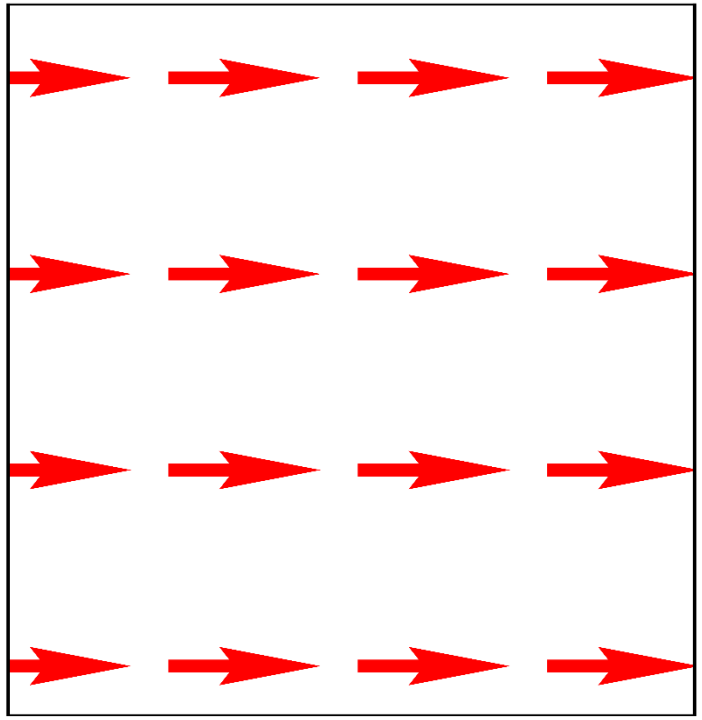
[Mikolajczyk & Matas. ICCV '07]

Shift invariant similarity

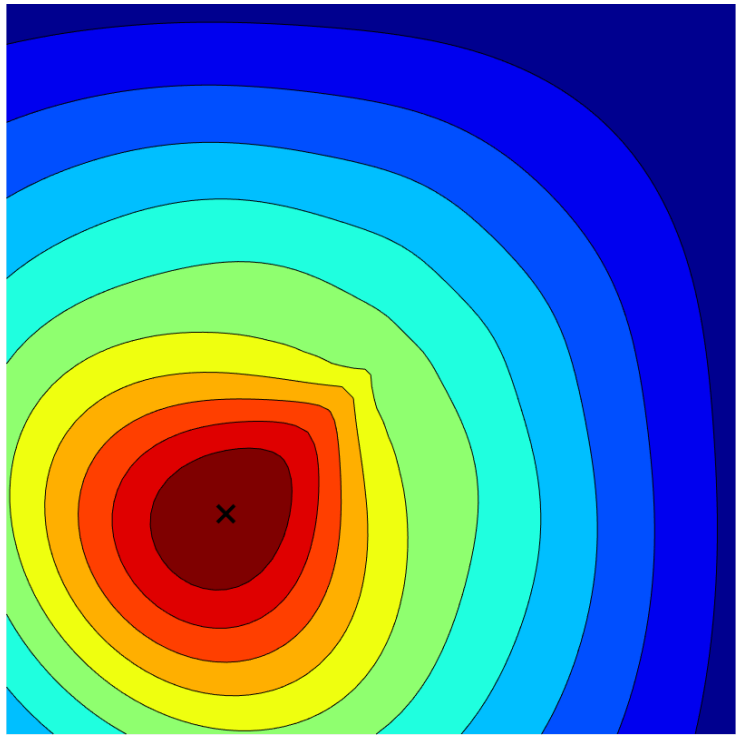
Stationary kernel: depends on difference of angles, not their absolute value



Pixel p

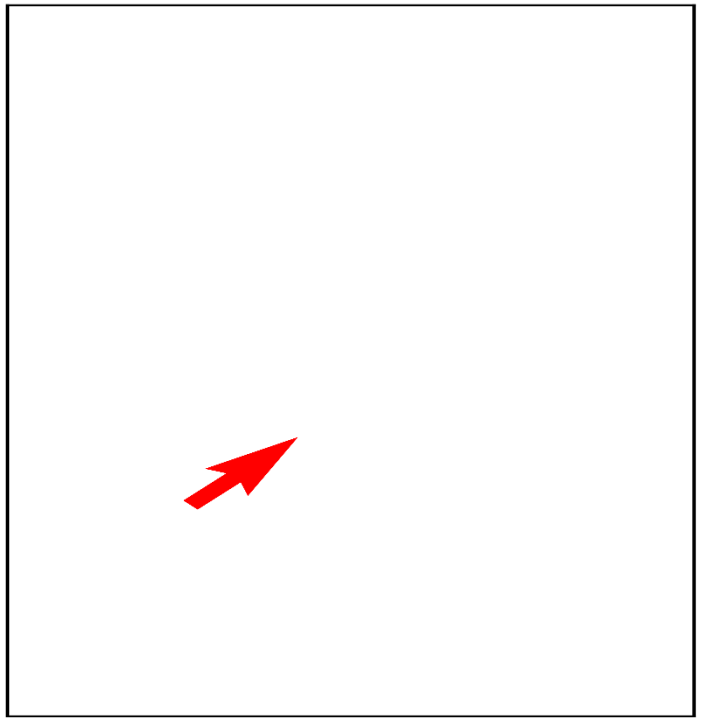


Patch $\mathcal{Q}$

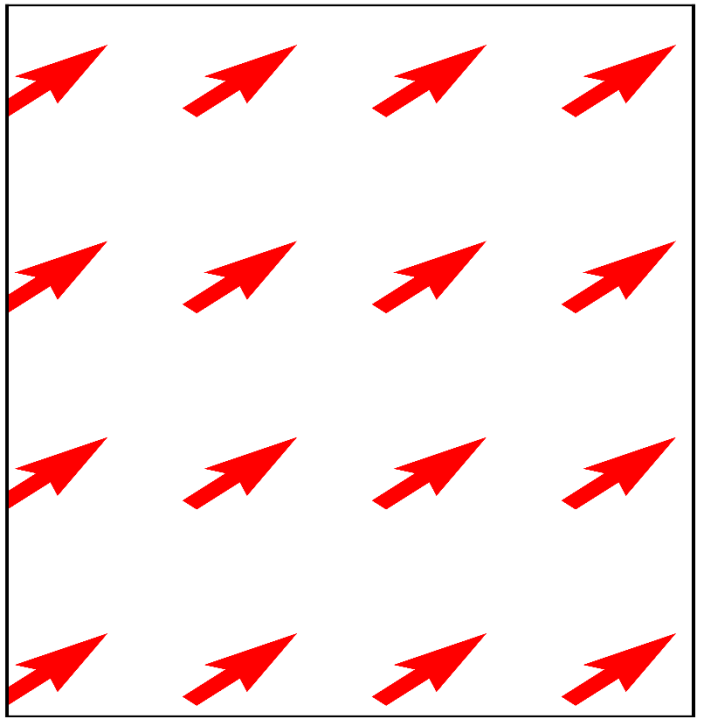


Shift invariant similarity

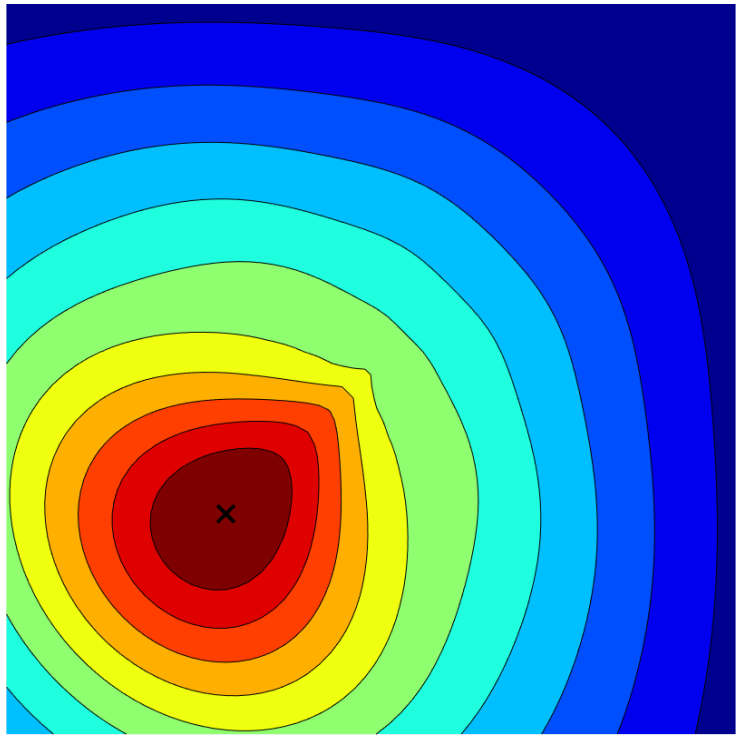
Stationary kernel: depends on difference of angles, not their absolute value



Pixel p

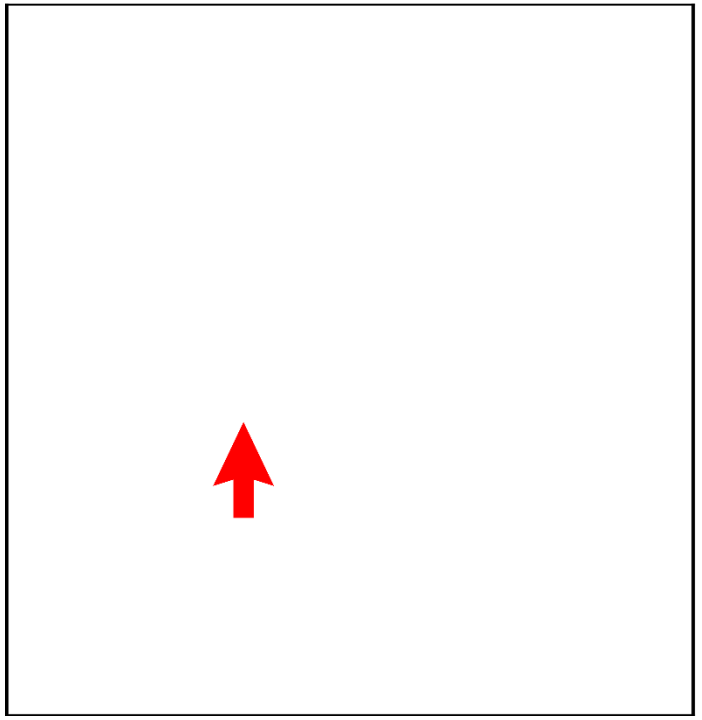


Patch $\mathcal{Q}$

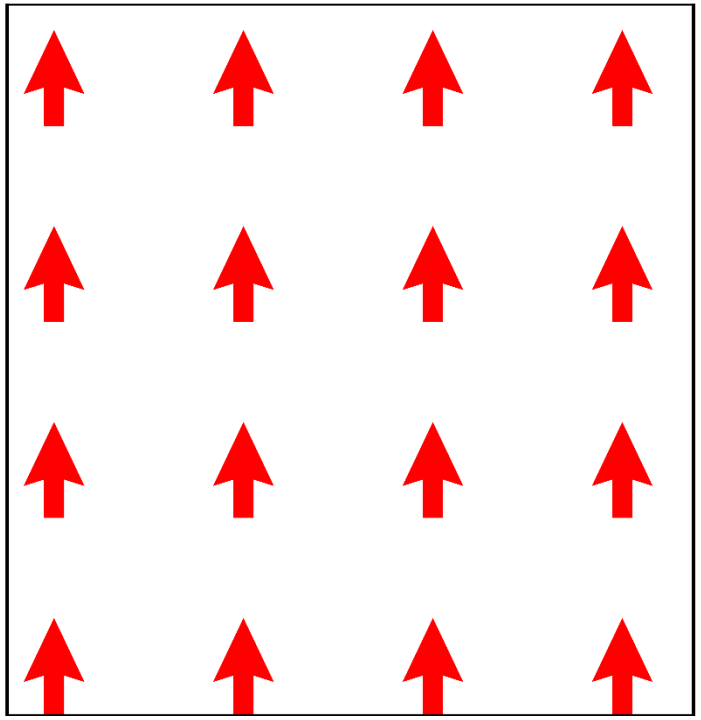


Shift invariant similarity

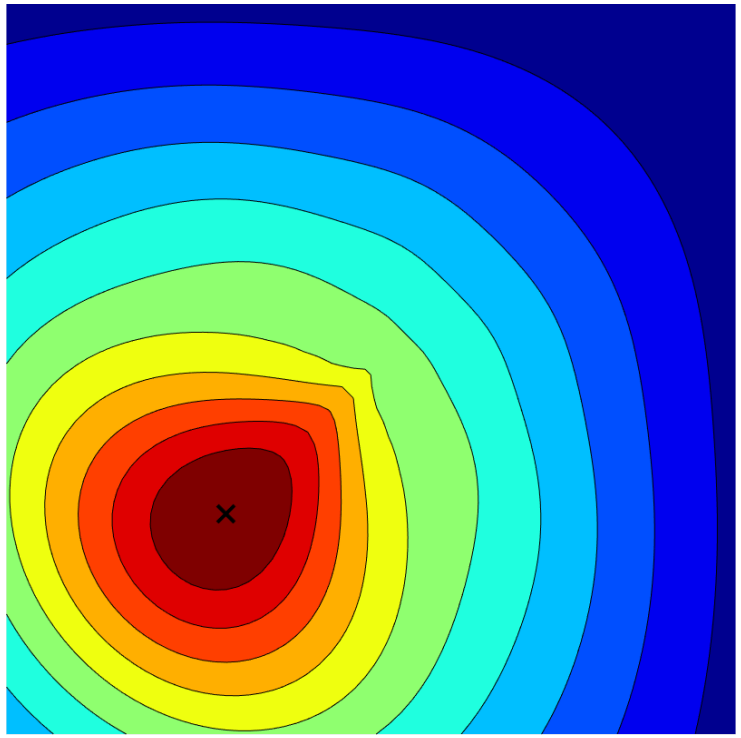
Stationary kernel: depends on difference of angles, not their absolute value



Pixel p

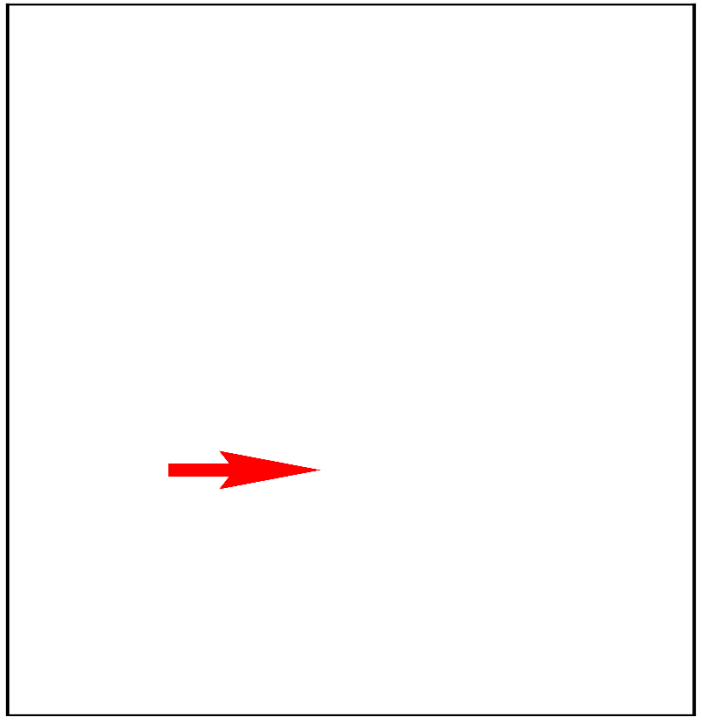


Patch $\mathcal{Q}$

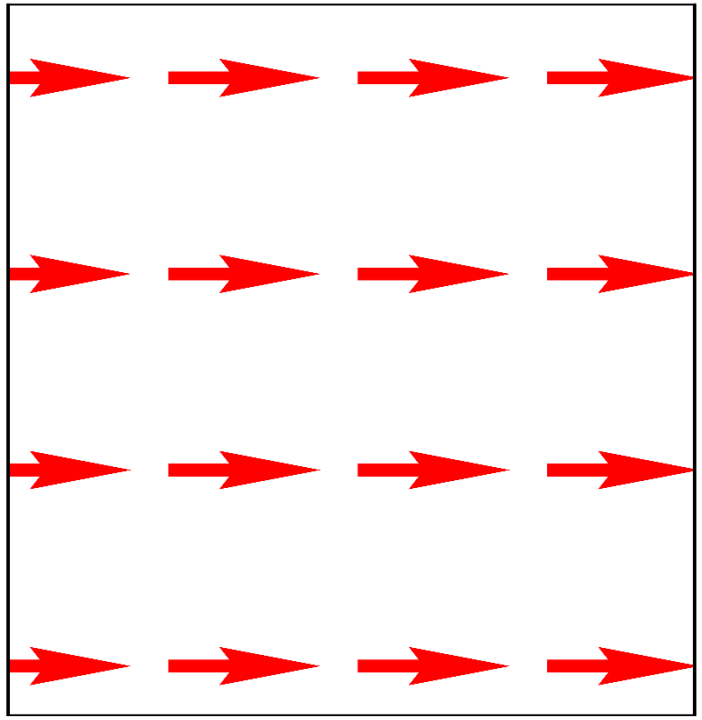


Similarity learned by supervised whitening

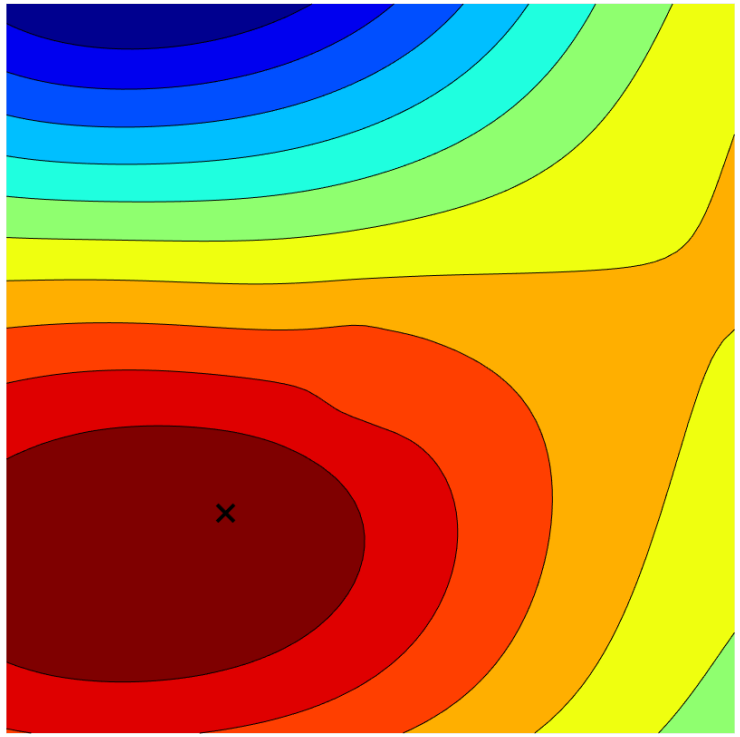
Similarity function is not shift invariant anymore!



Pixel p

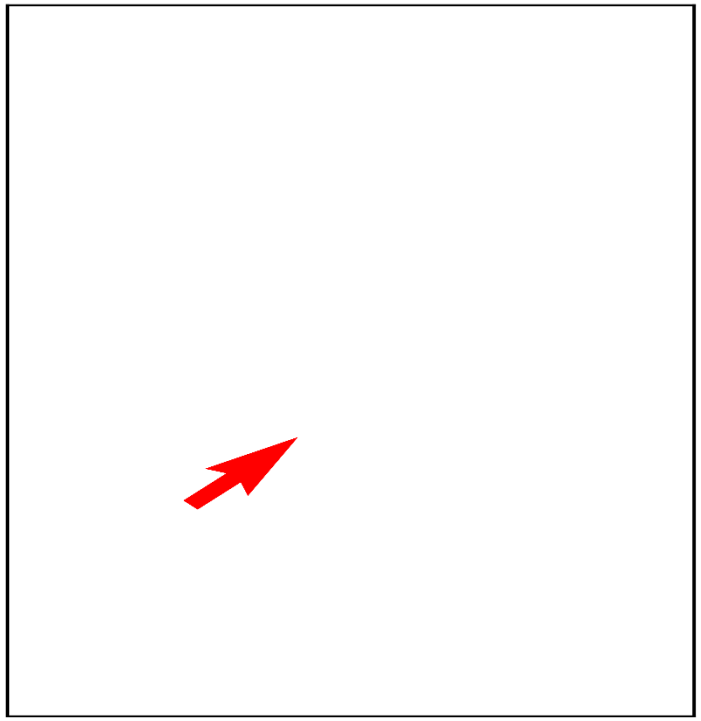


Patch $\mathcal{Q}$

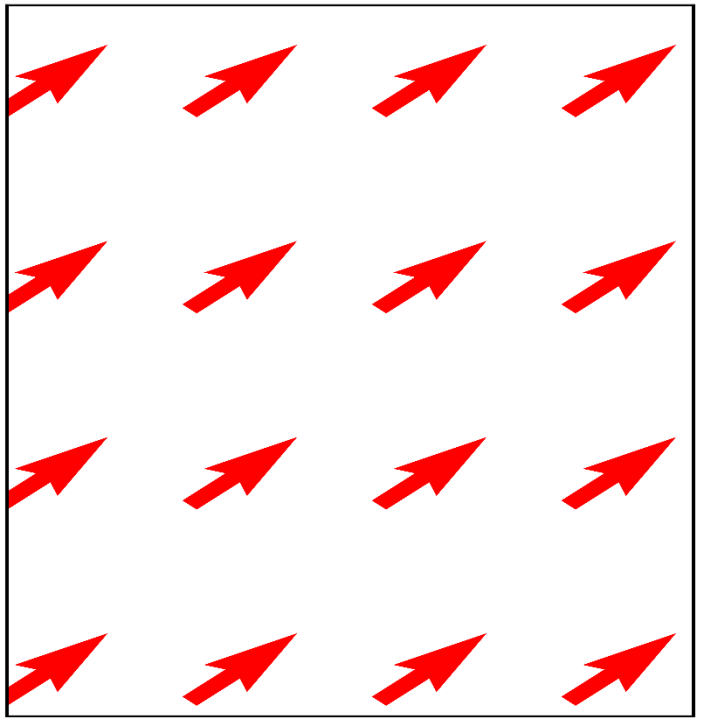


Similarity learned by supervised whitening

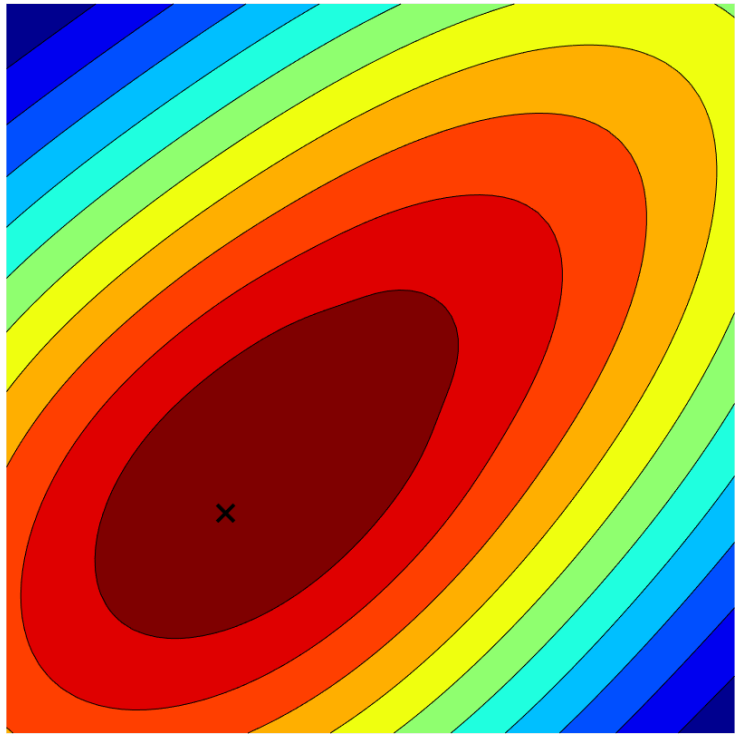
Similarity function is not shift invariant anymore!



Pixel p

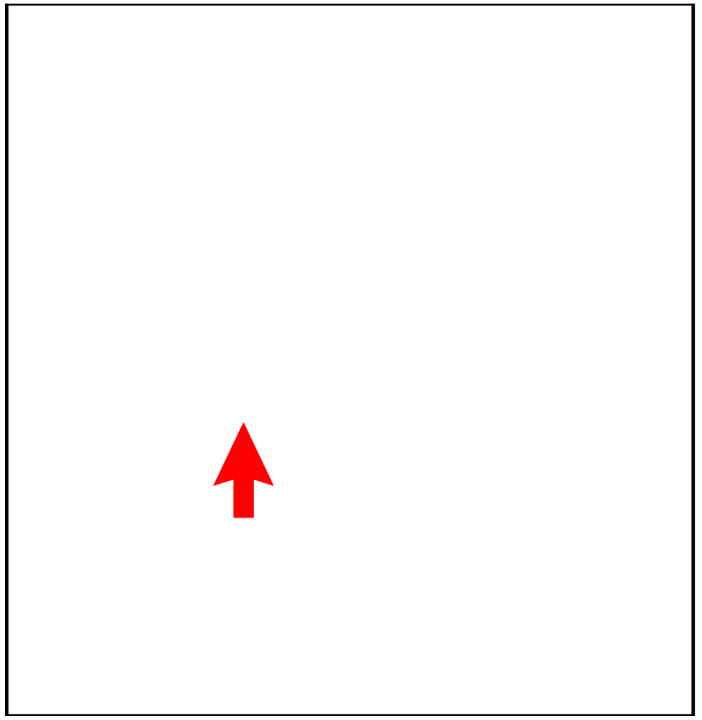


Patch $\mathcal{Q}$

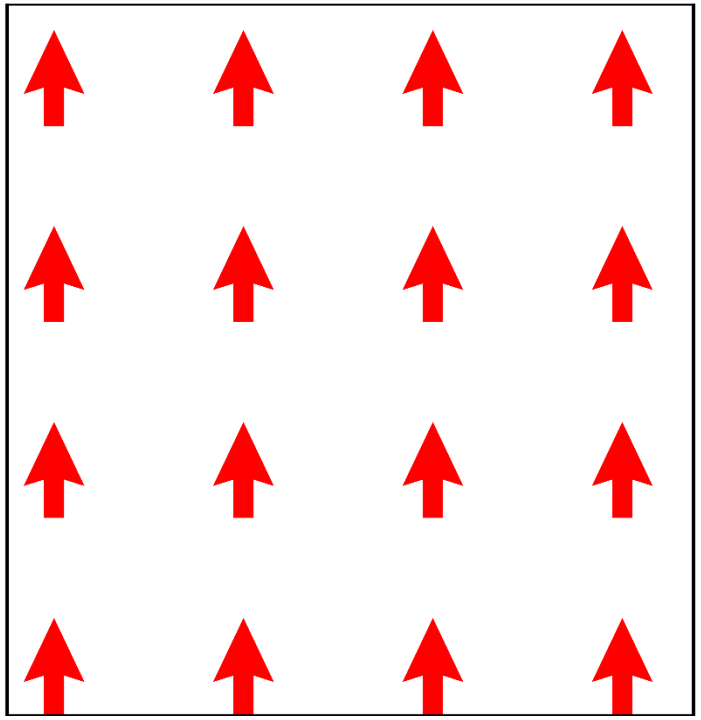


Similarity learned by supervised whitening

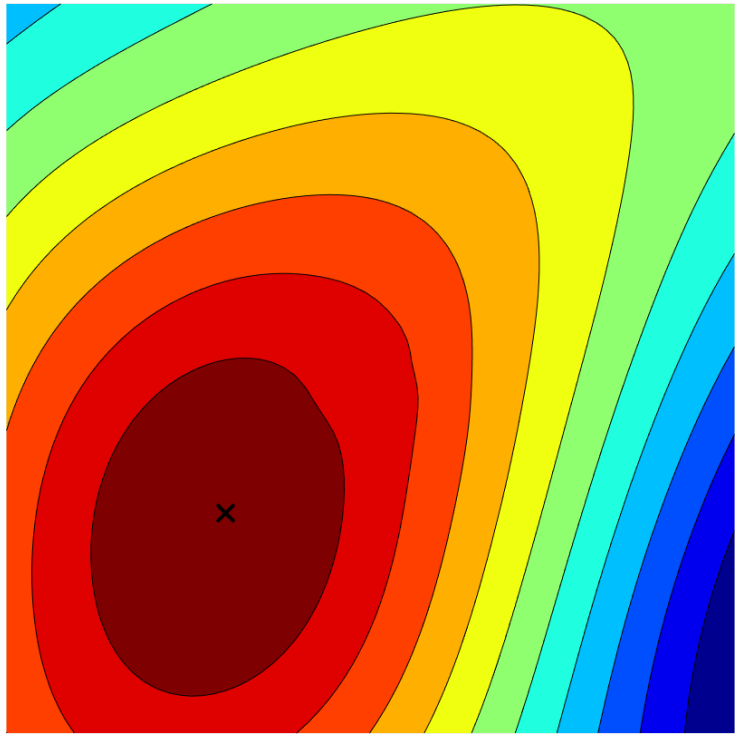
Similarity function is not shift invariant anymore!



Pixel p

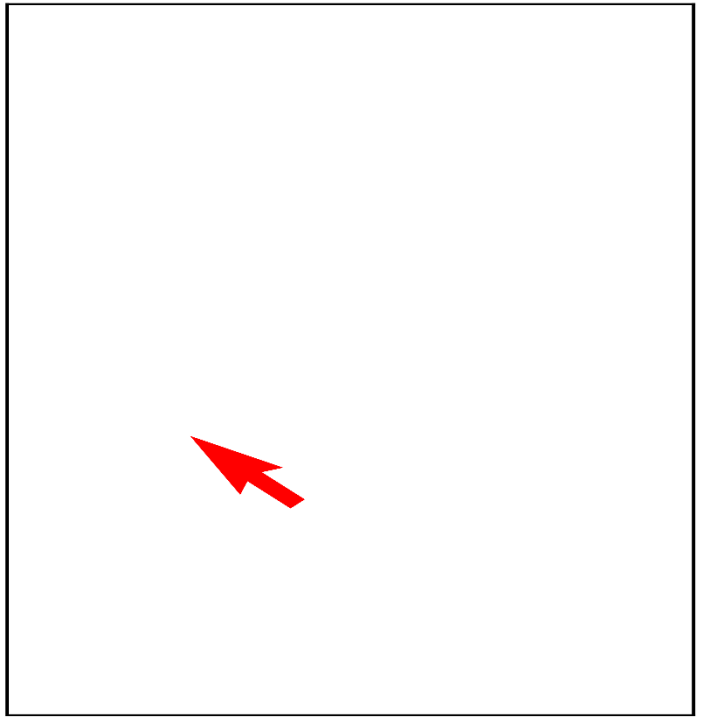


Patch $\mathcal{Q}$

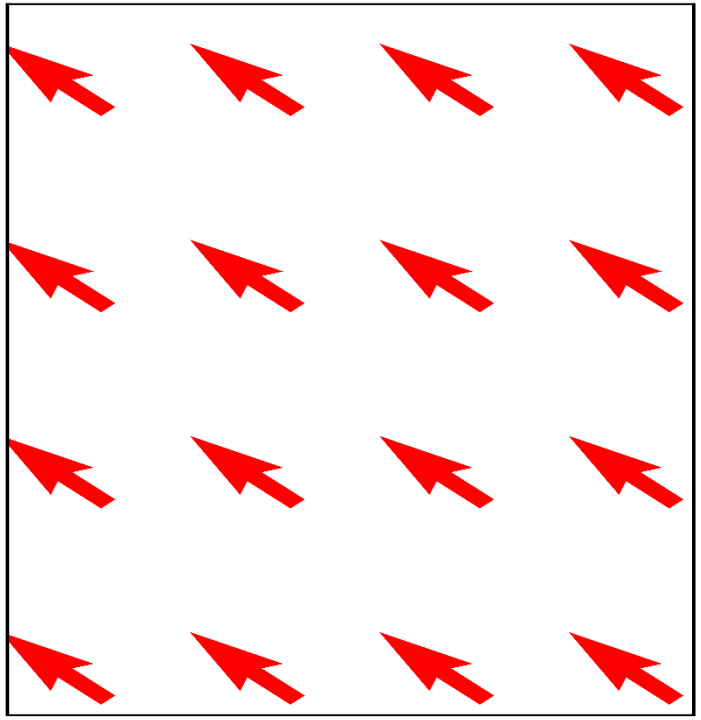


Similarity learned by supervised whitening

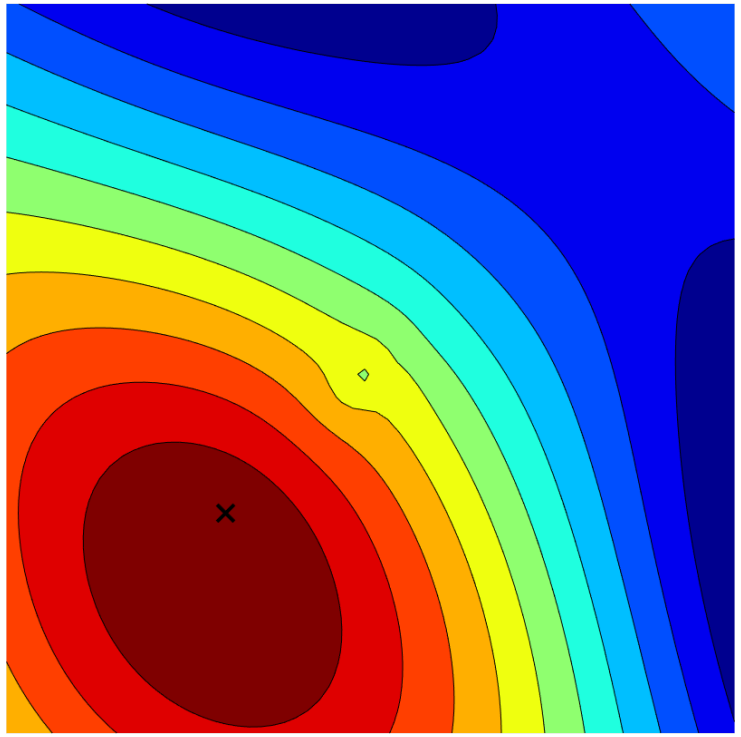
Similarity function is not shift invariant anymore!



Pixel p

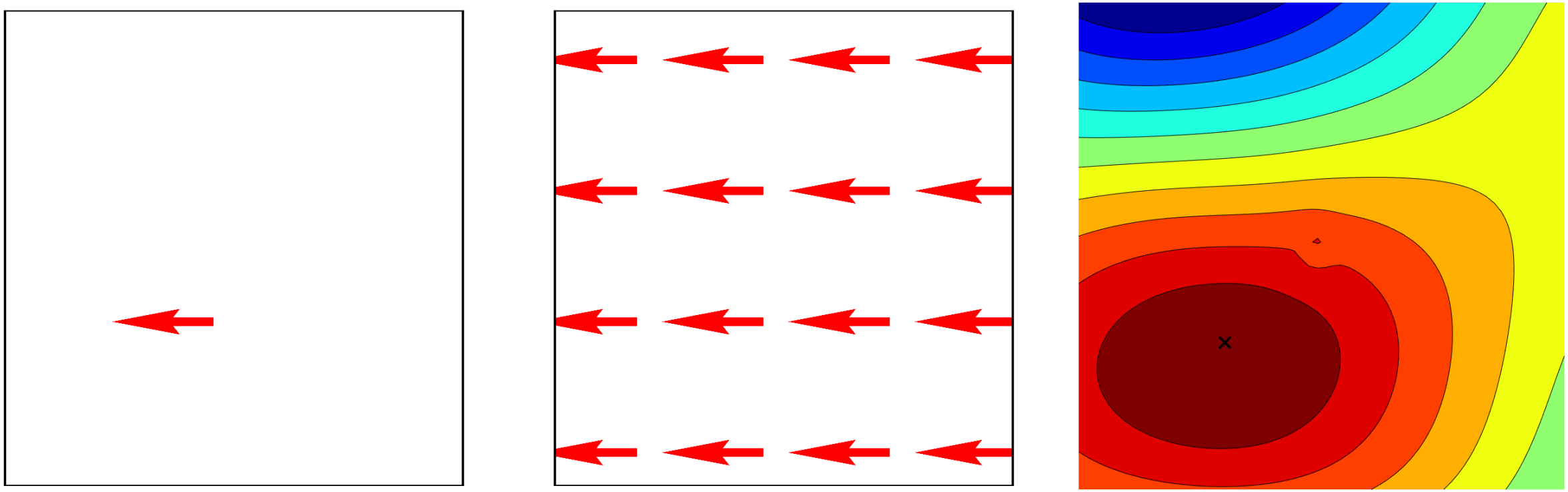


Patch $\mathcal{Q}$



Similarity learned by supervised whitening

Similarity function is not shift invariant anymore!

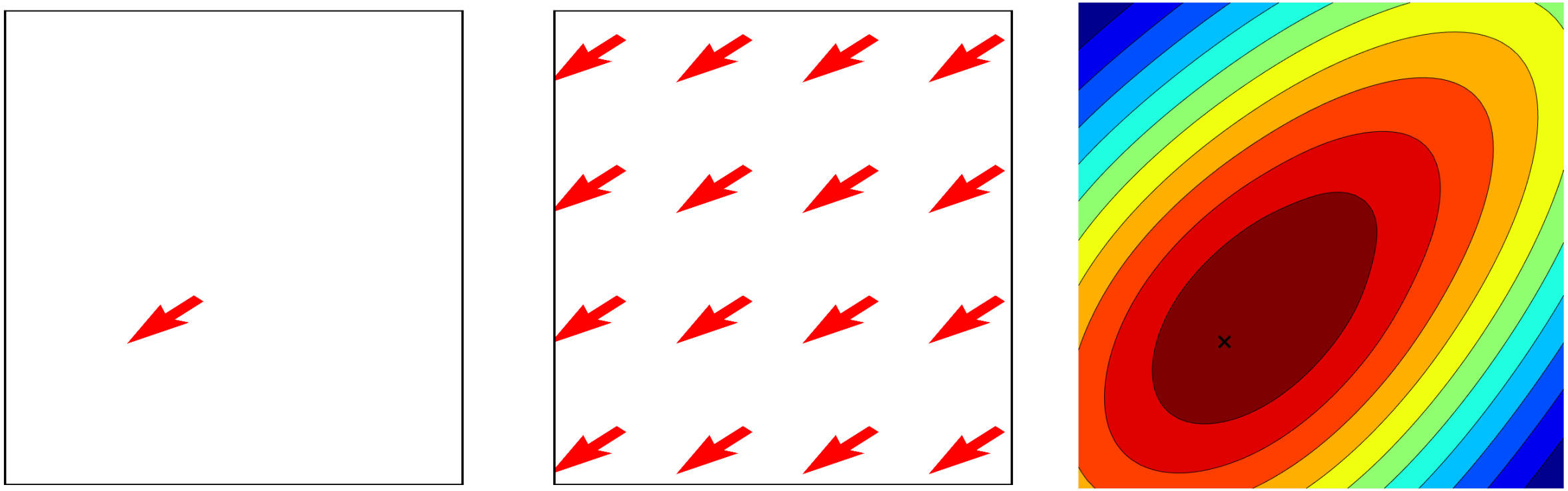


Pixel p

Patch $\mathcal{Q}$

Similarity learned by supervised whitening

Similarity function is not shift invariant anymore!

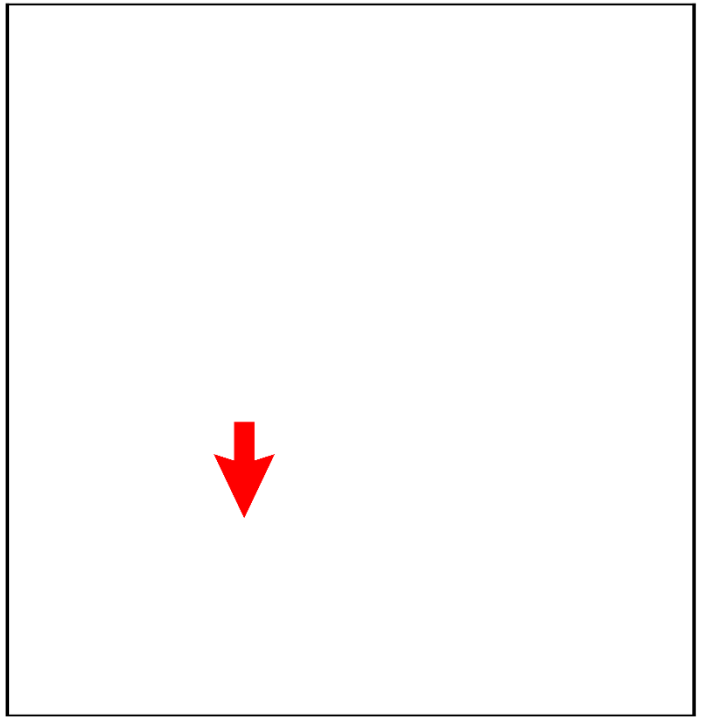


Pixel p

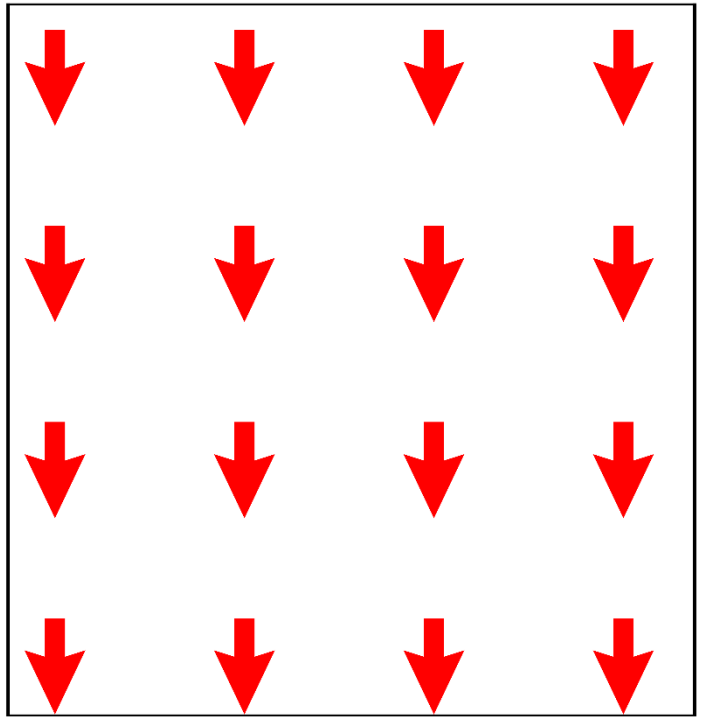
Patch $\mathcal{Q}$

Similarity learned by supervised whitening

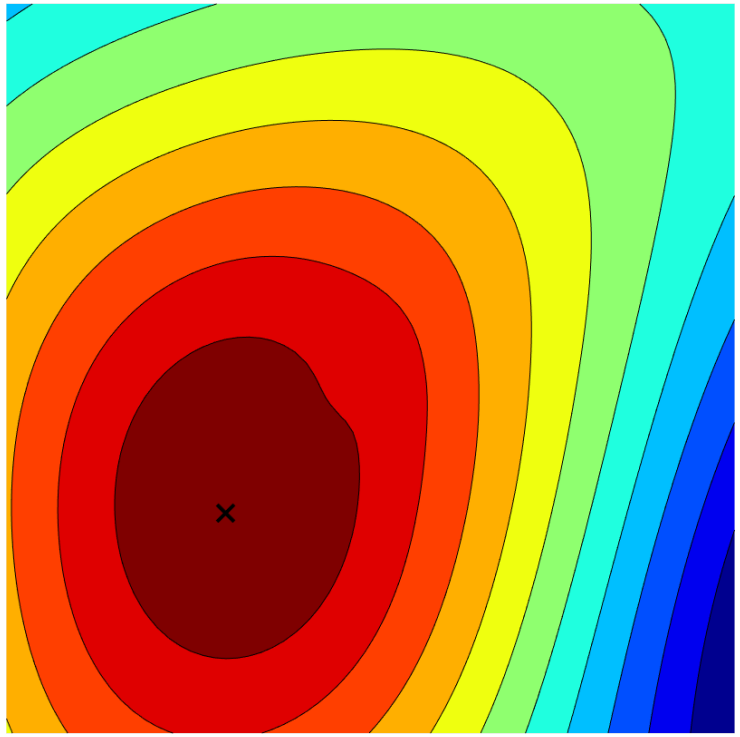
Similarity function is not shift invariant anymore!



Pixel p

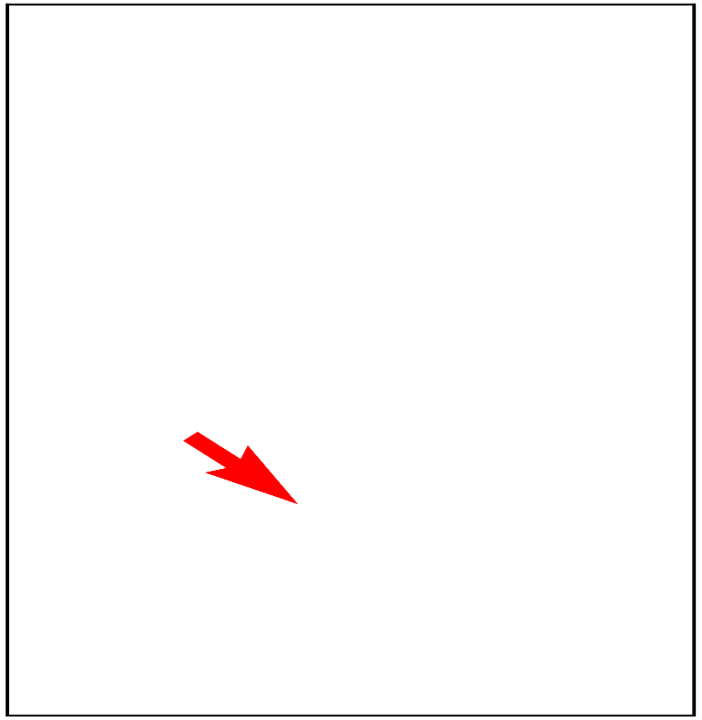


Patch $\mathcal{Q}$

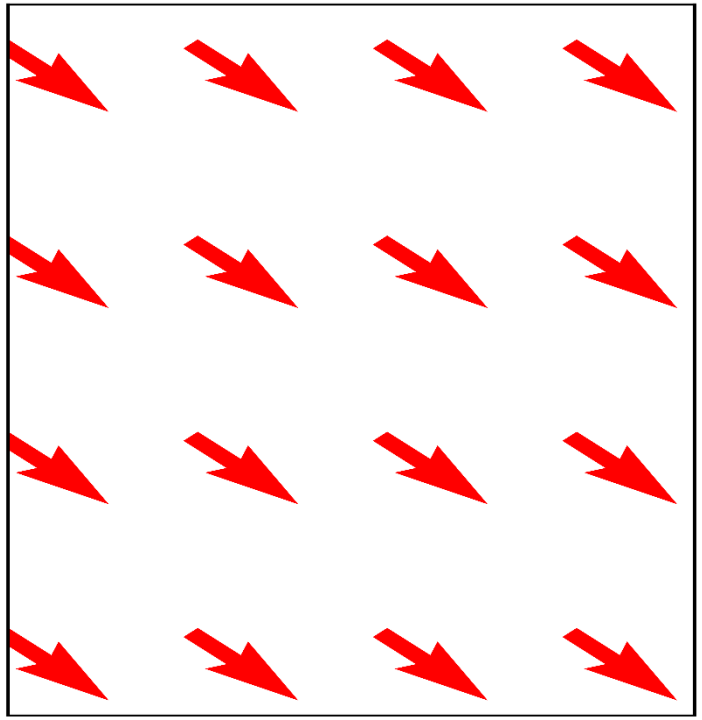


Similarity learned by supervised whitening

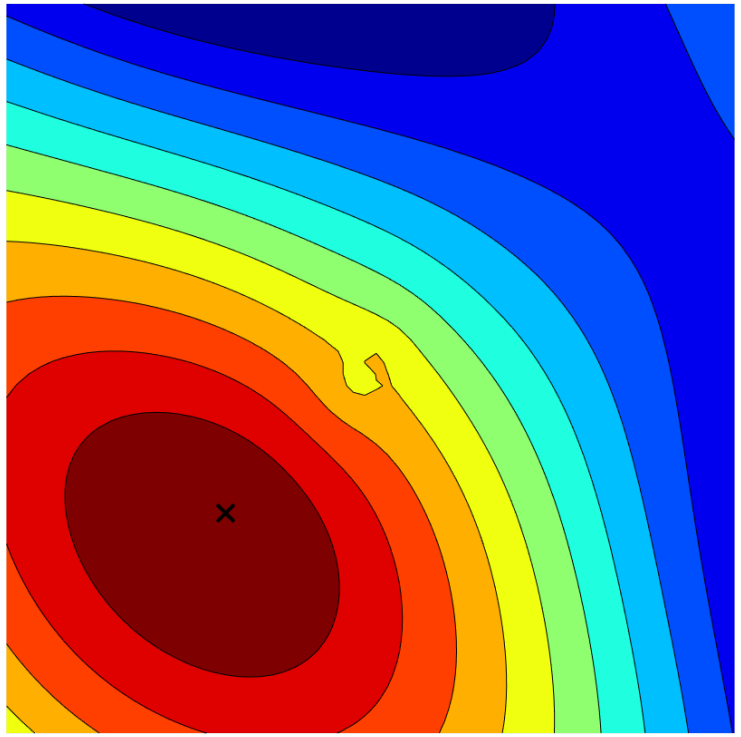
Similarity function is not shift invariant anymore!



Pixel p

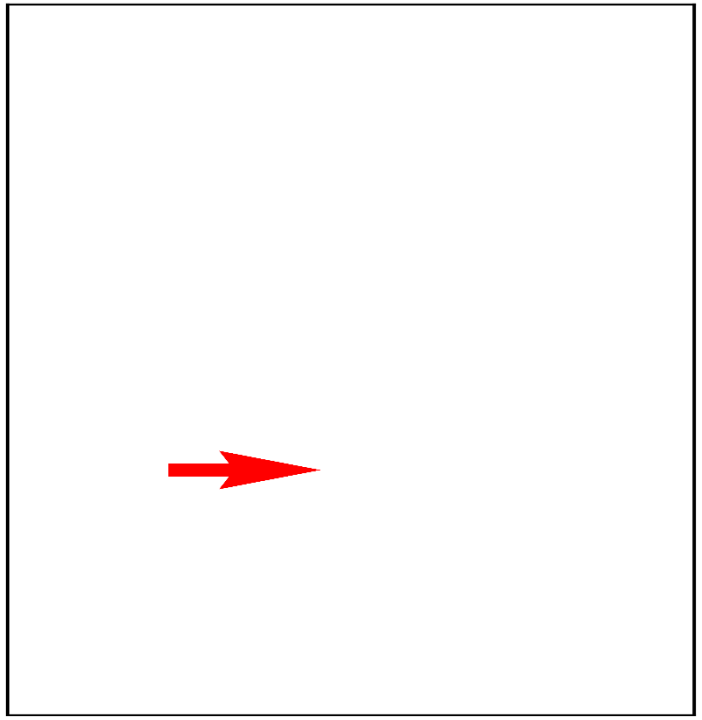


Patch $\mathcal{Q}$

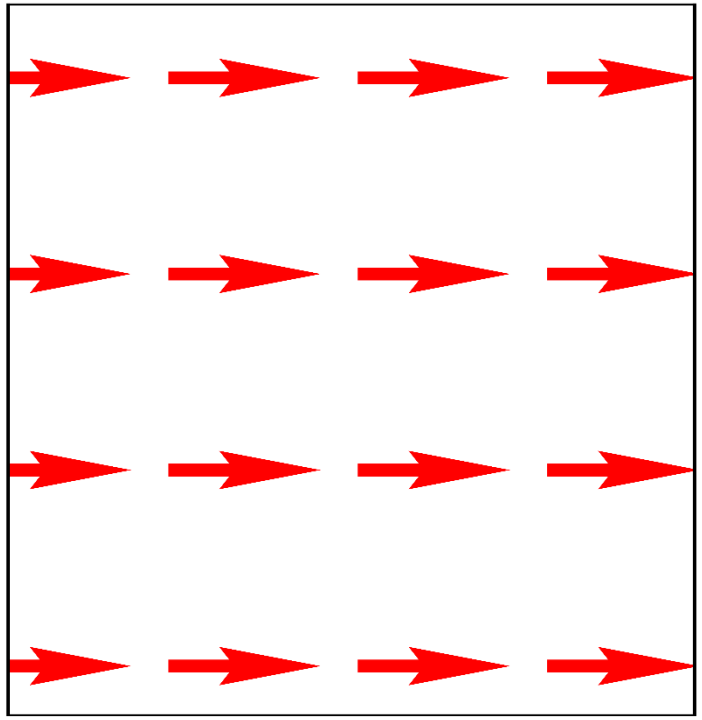


Similarity learned by supervised whitening

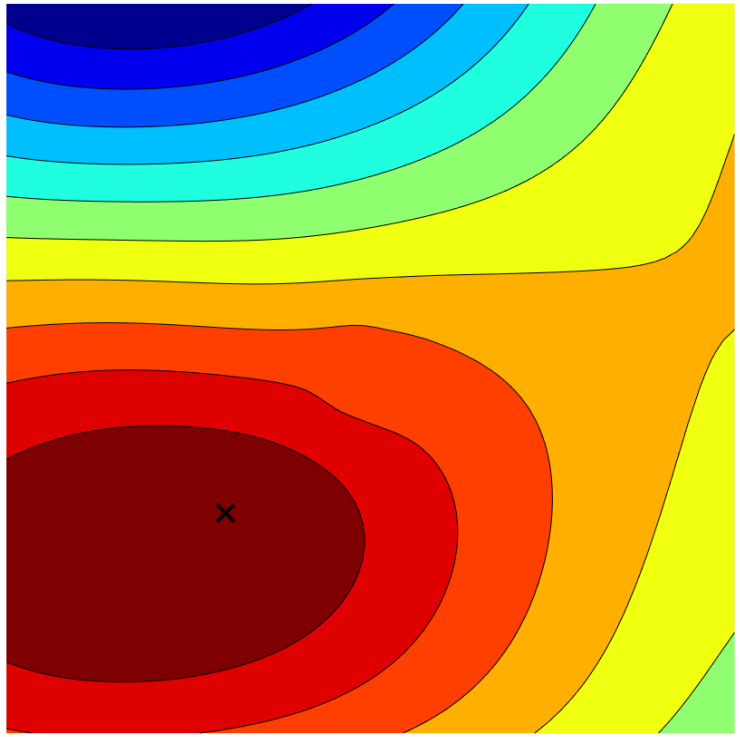
Similarity function is not shift invariant anymore!



Pixel p

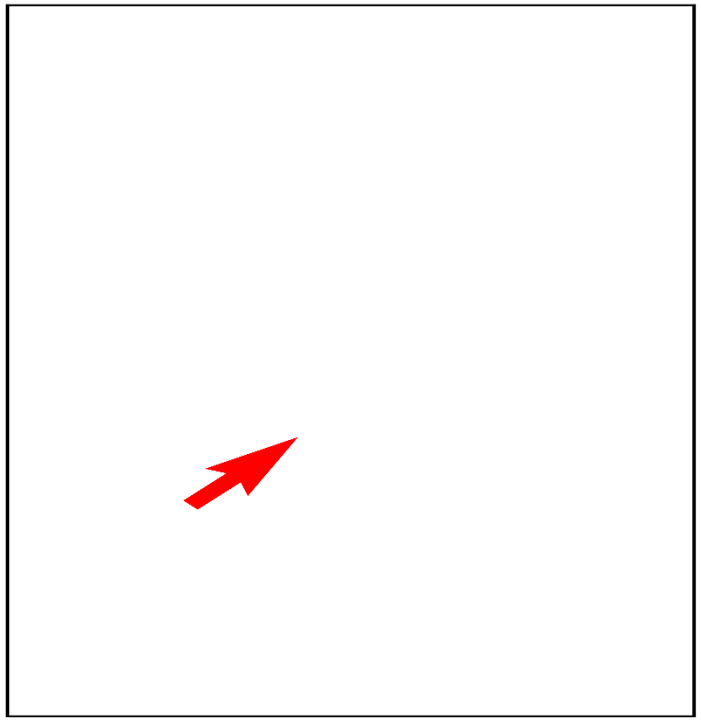


Patch $\mathcal{Q}$

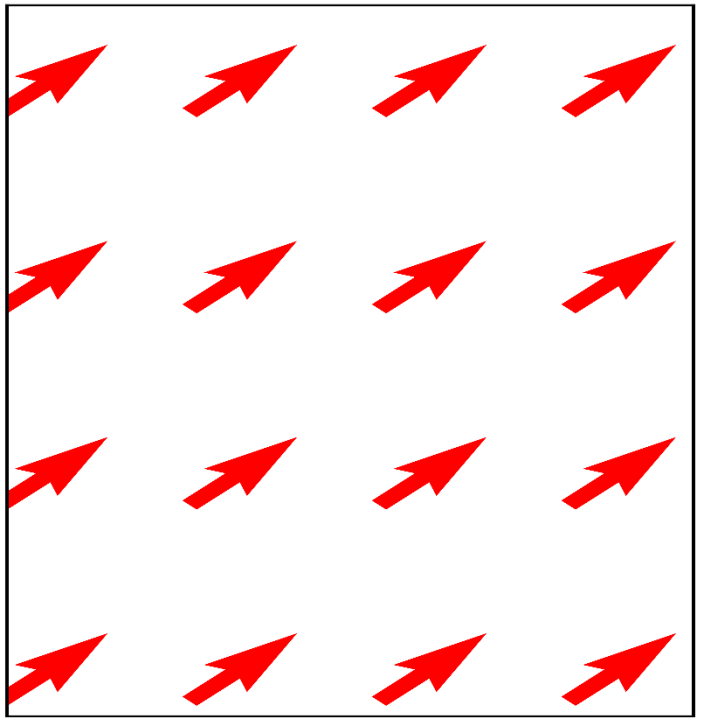


Similarity learned by supervised whitening

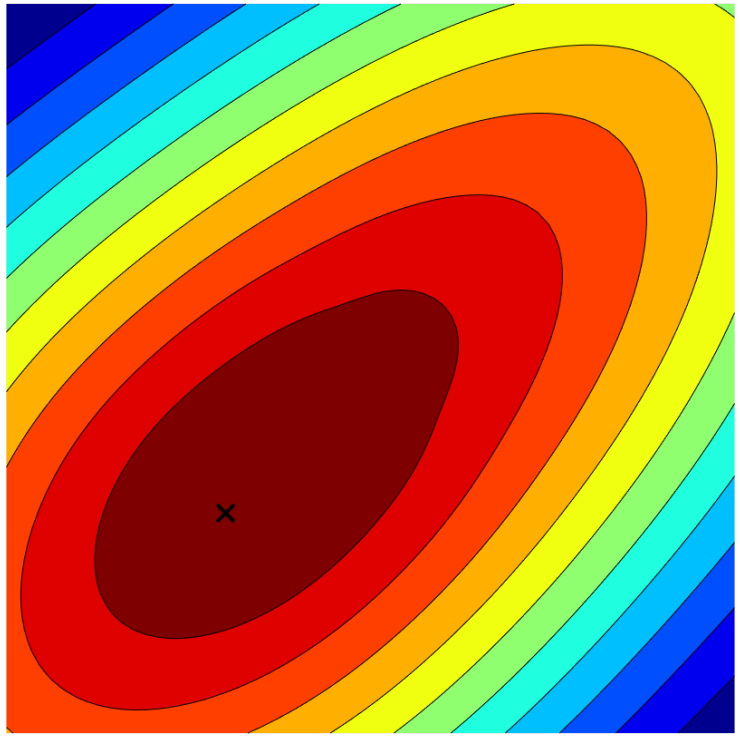
Similarity function is not shift invariant anymore!



Pixel p

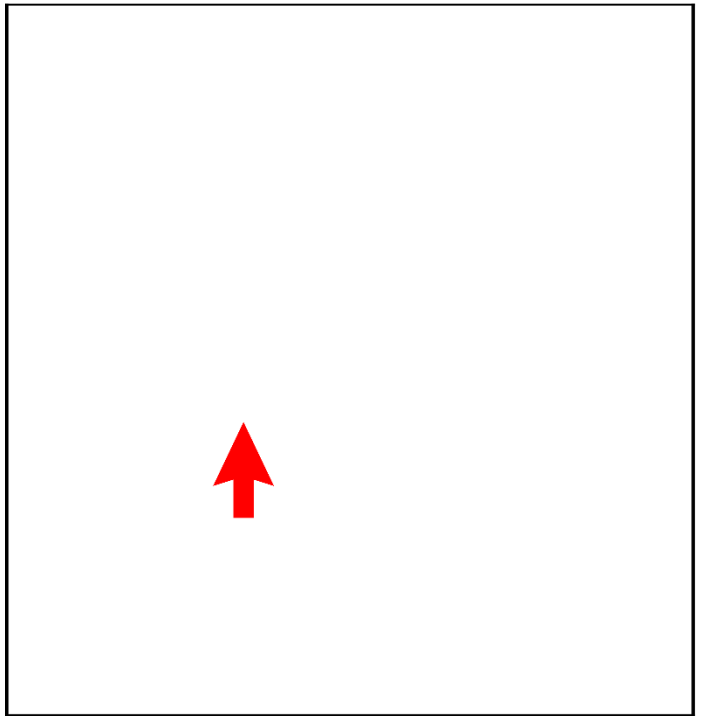


Patch $\mathcal{Q}$

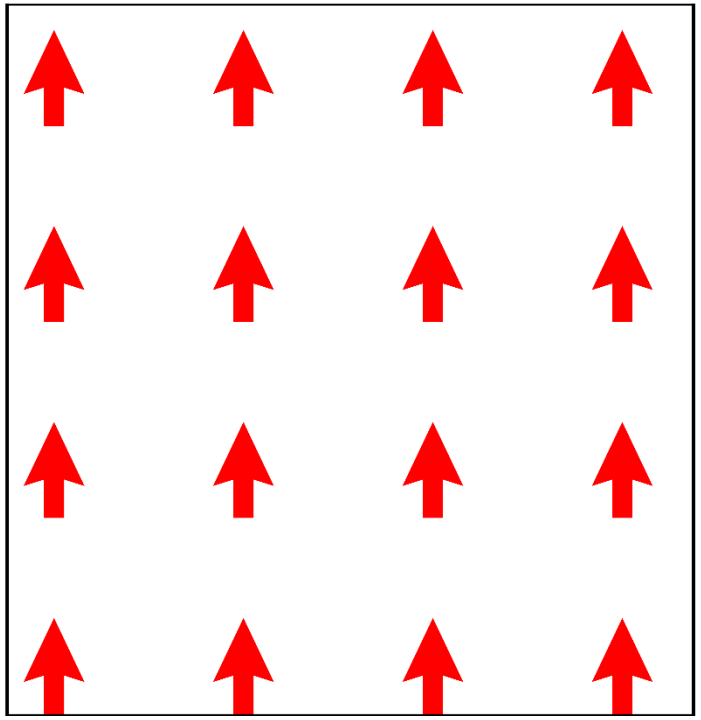


Similarity learned by supervised whitening

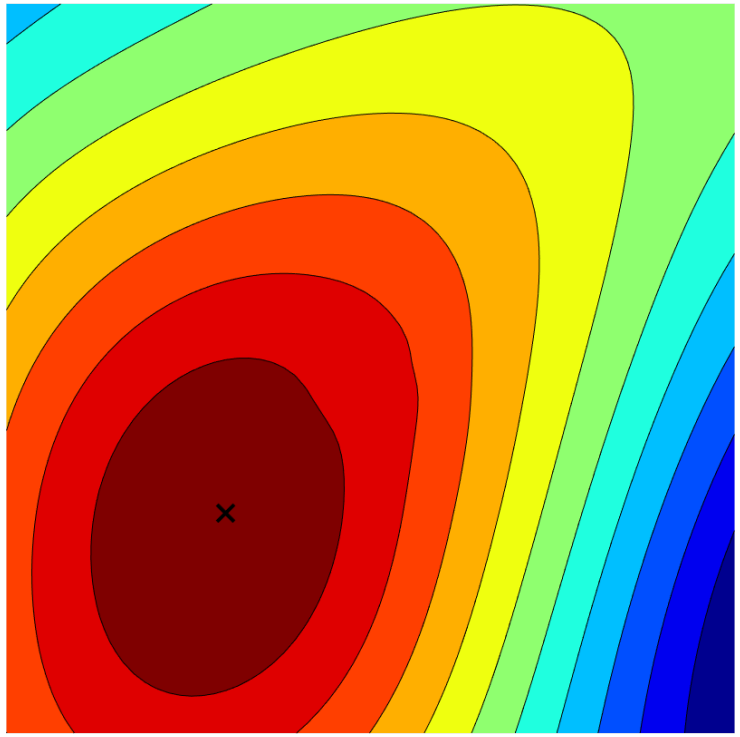
Similarity function is not shift invariant anymore!



Pixel p

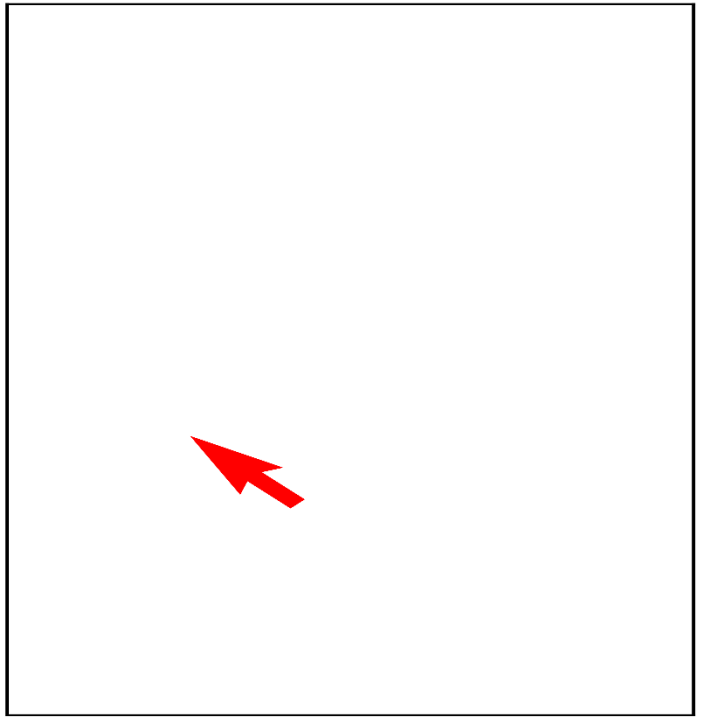


Patch $\mathcal{Q}$

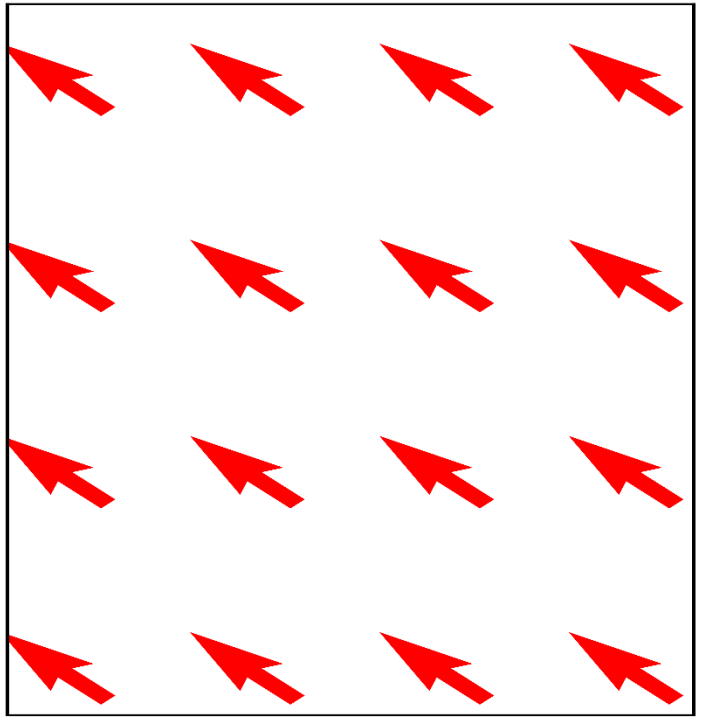


Similarity learned by supervised whitening

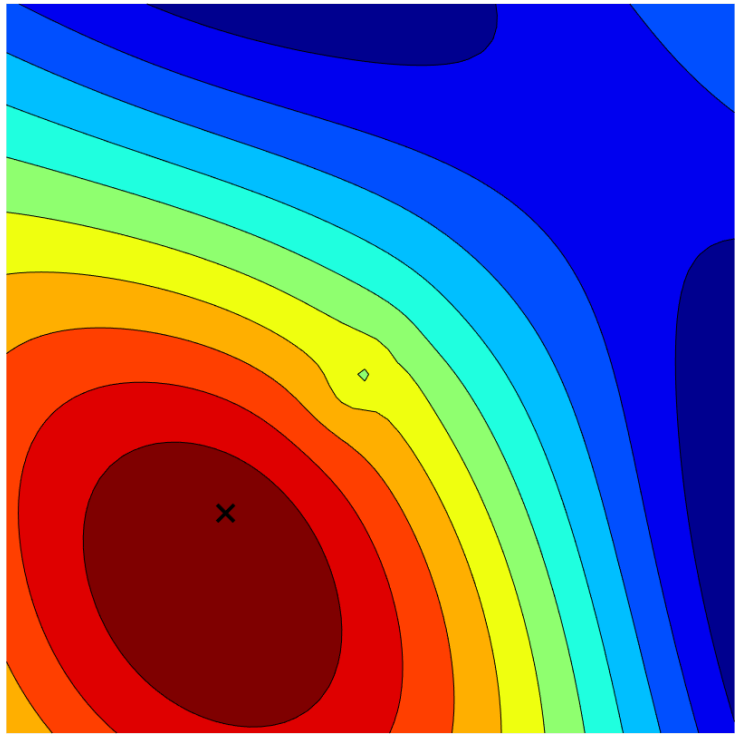
Similarity function is not shift invariant anymore!



Pixel p

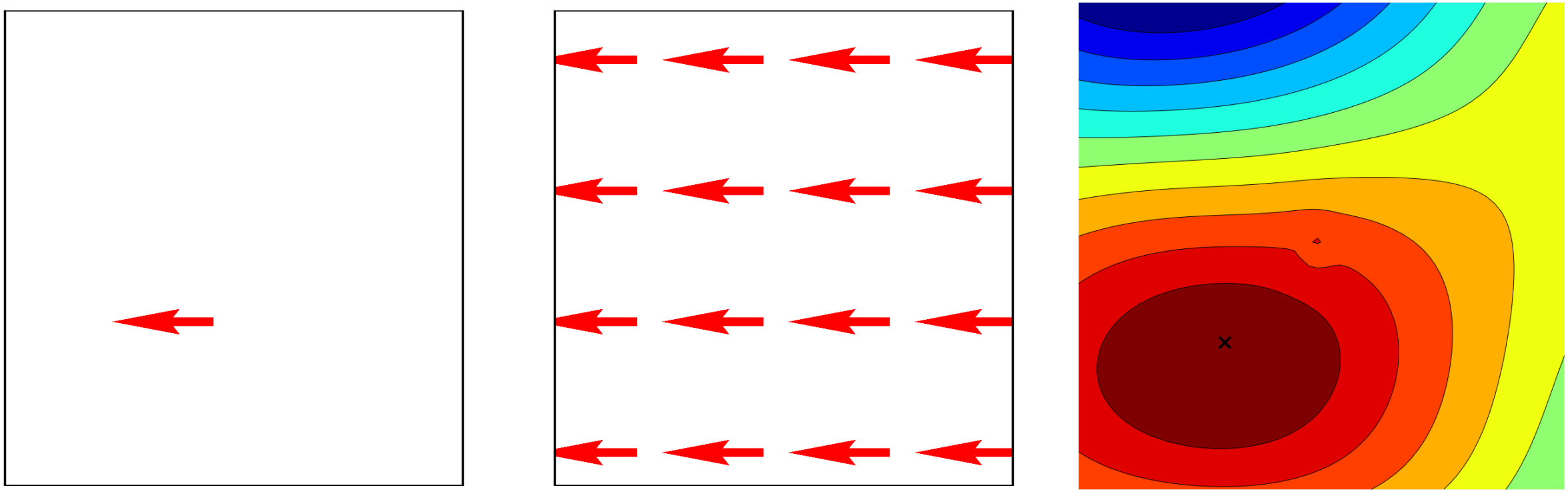


Patch $\mathcal{Q}$



Similarity learned by supervised whitening

Similarity function is not shift invariant anymore!

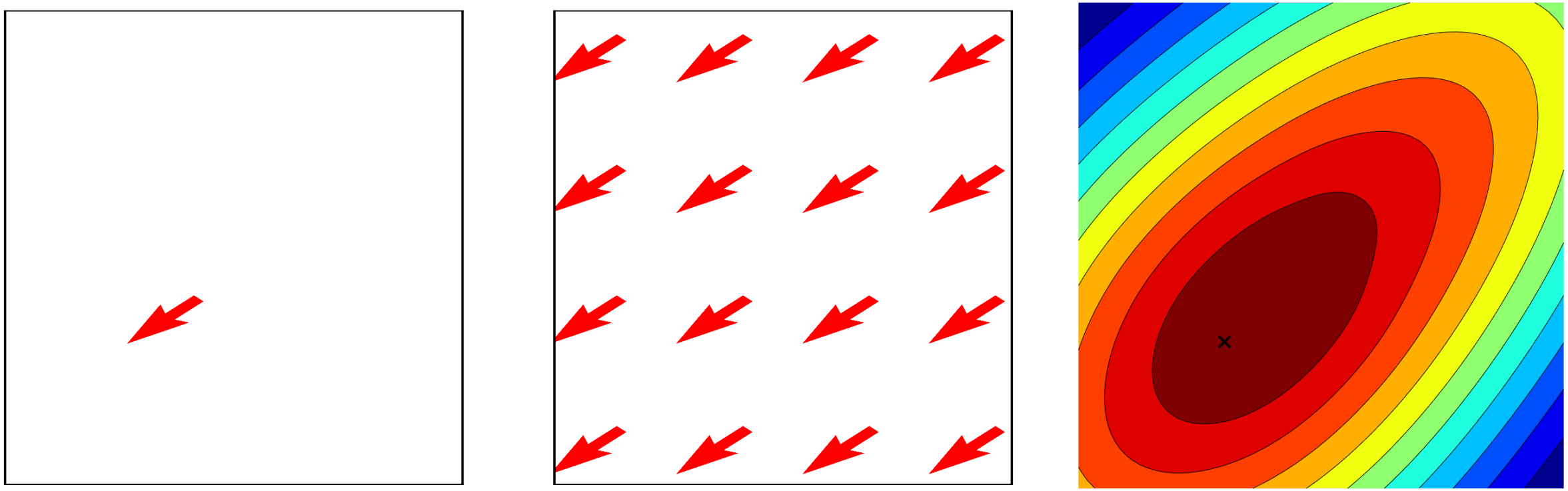


Pixel p

Patch $\mathcal{Q}$

Similarity learned by supervised whitening

Similarity function is not shift invariant anymore!

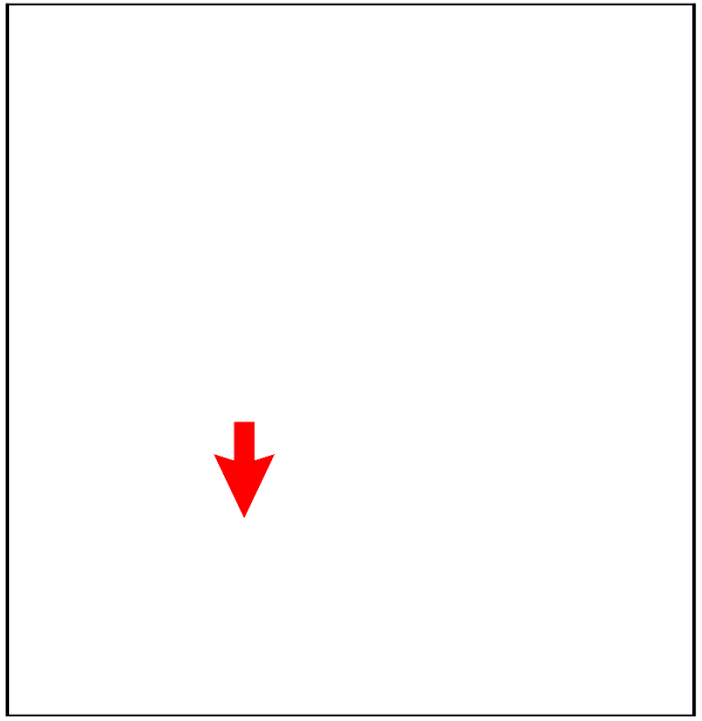


Pixel p

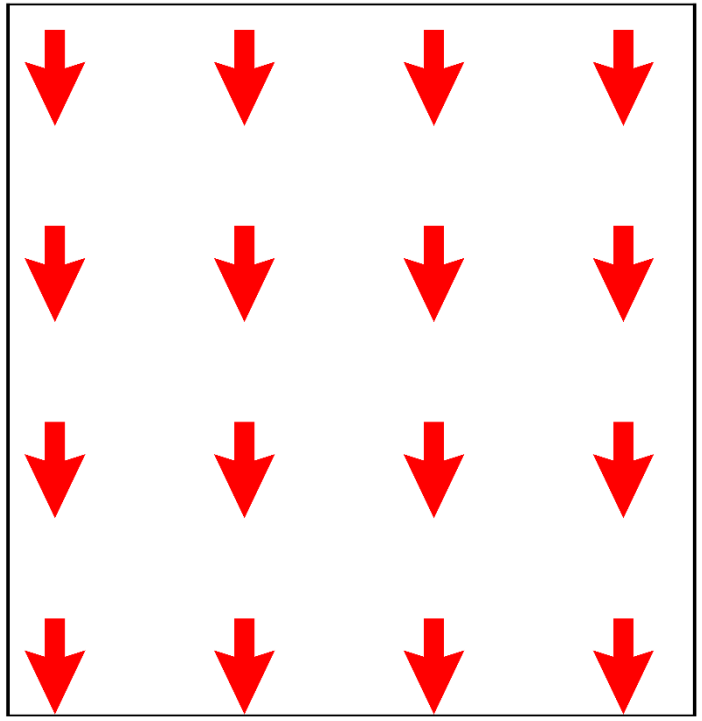
Patch $\mathcal{Q}$

Similarity learned by supervised whitening

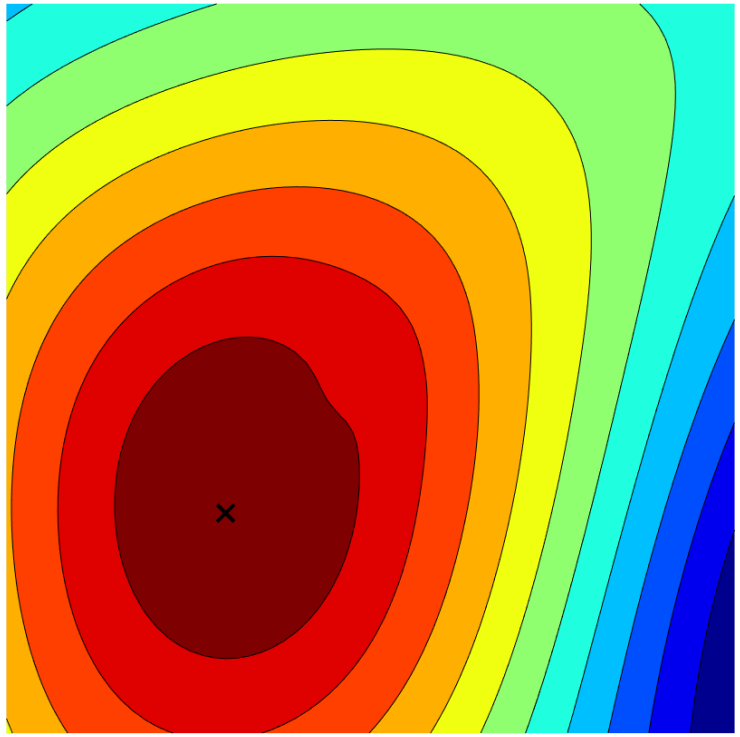
Similarity function is not shift invariant anymore!



Pixel p

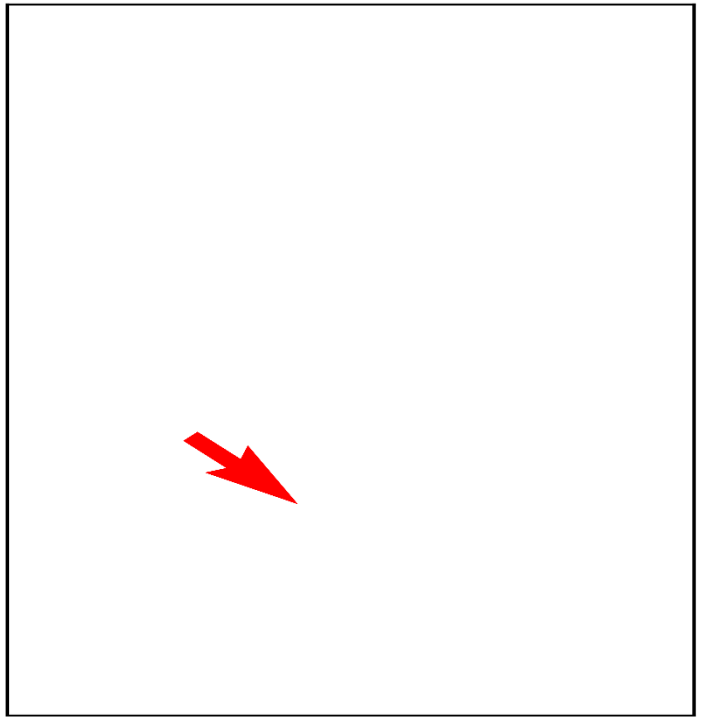


Patch $\mathcal{Q}$

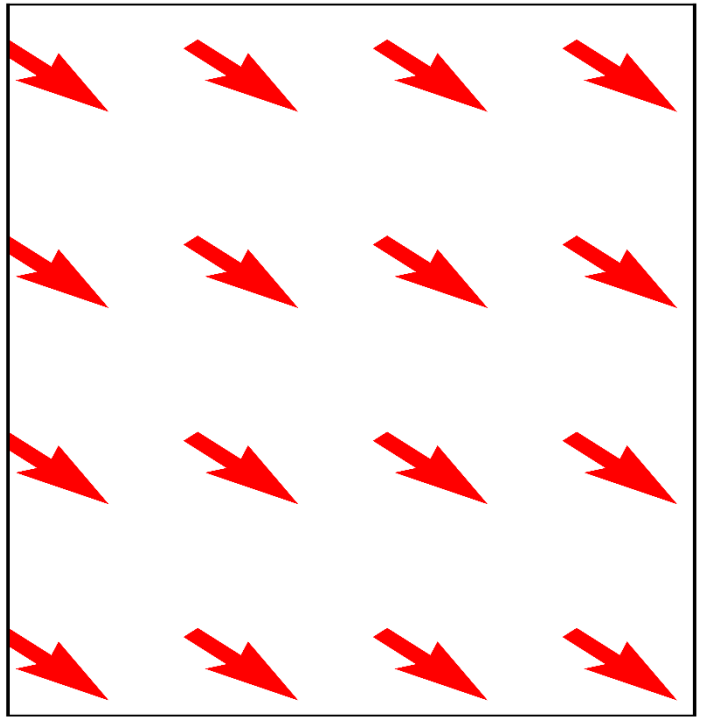


Similarity learned by supervised whitening

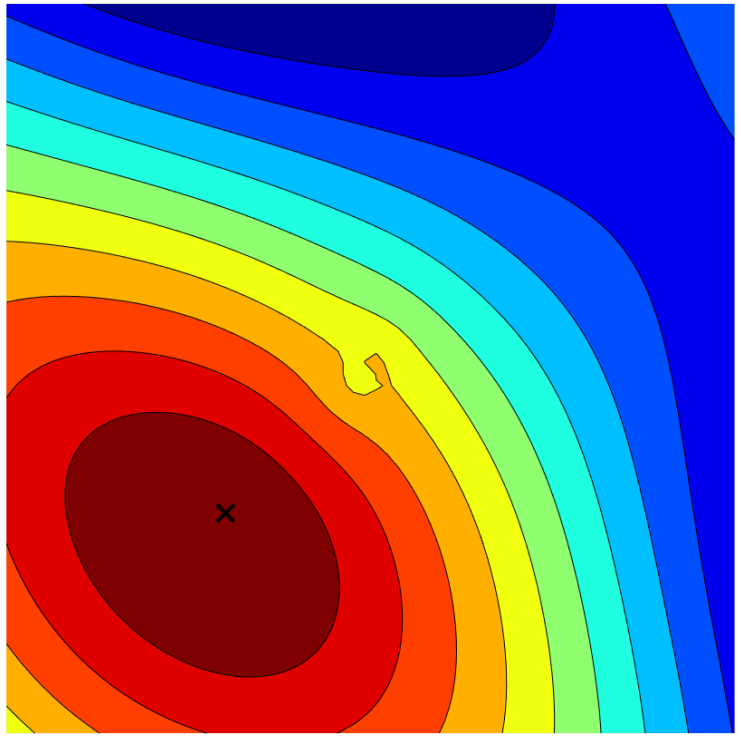
Similarity function is not shift invariant anymore!



Pixel p

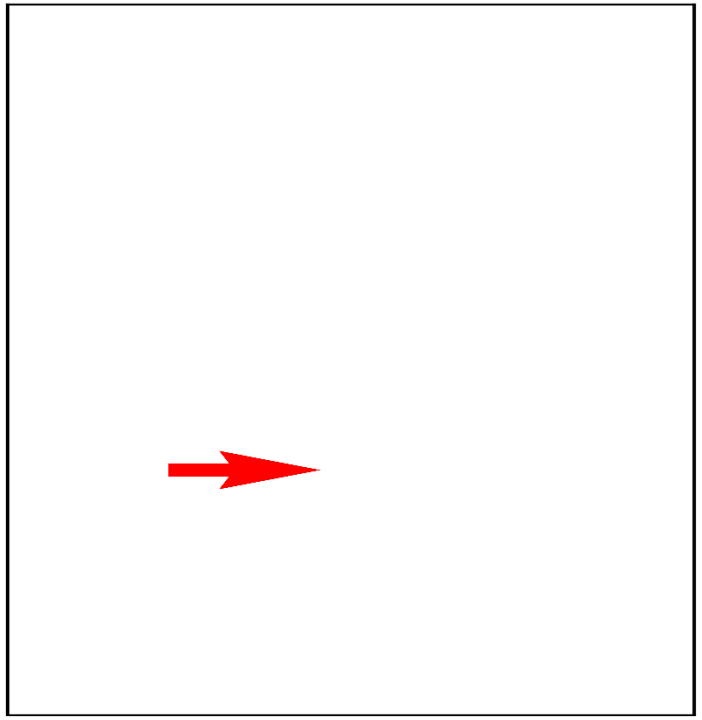


Patch $\mathcal{Q}$

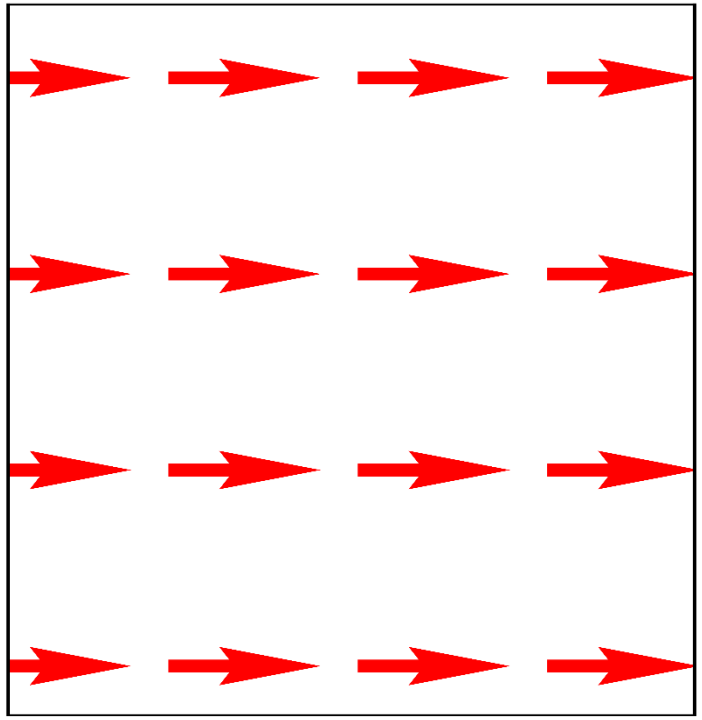


Similarity learned by supervised whitening

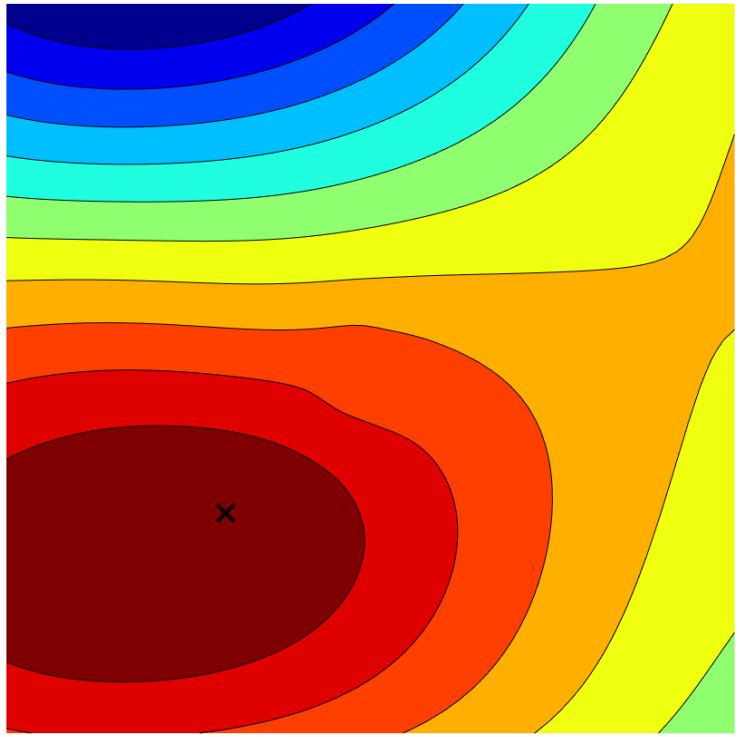
Similarity function is not shift invariant anymore!



Pixel p

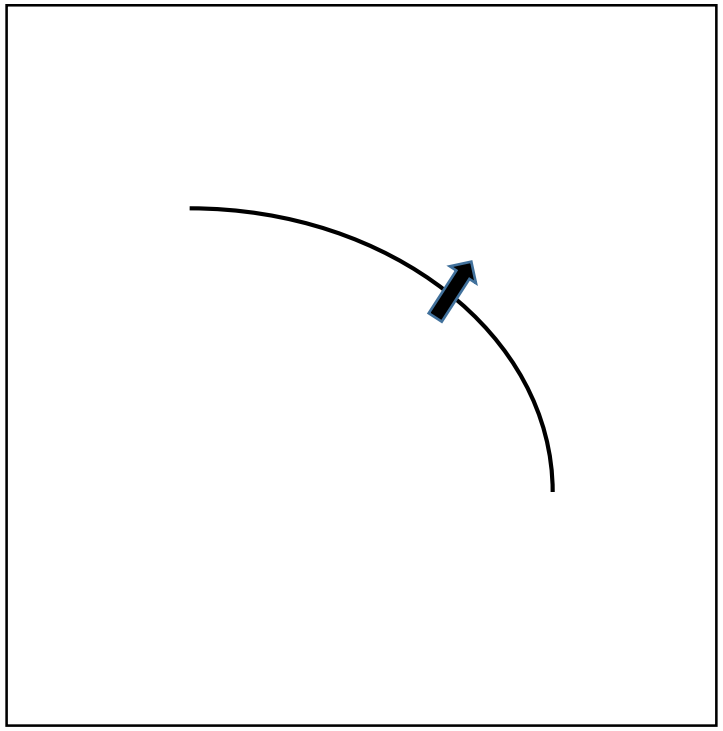


Patch $\mathcal{Q}$

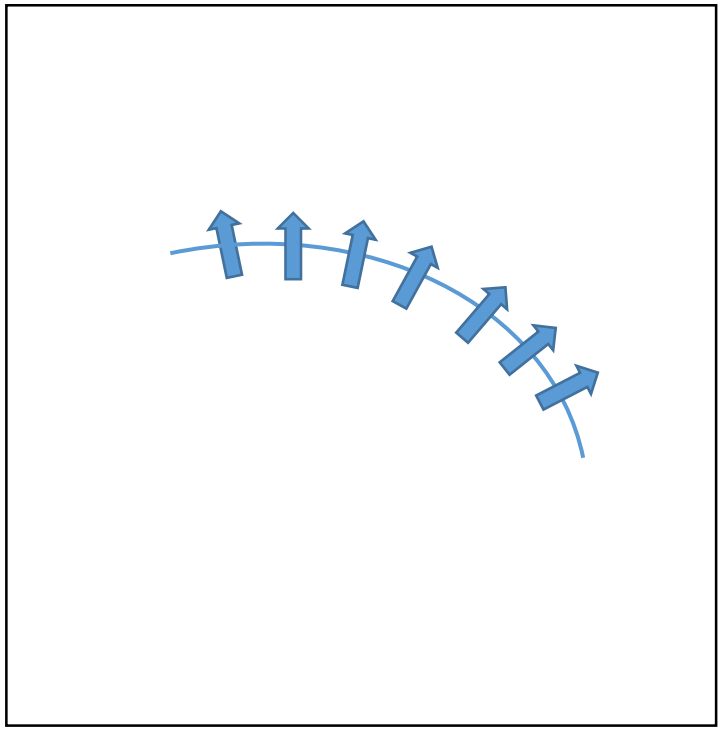


Similarity learned by supervised whitening

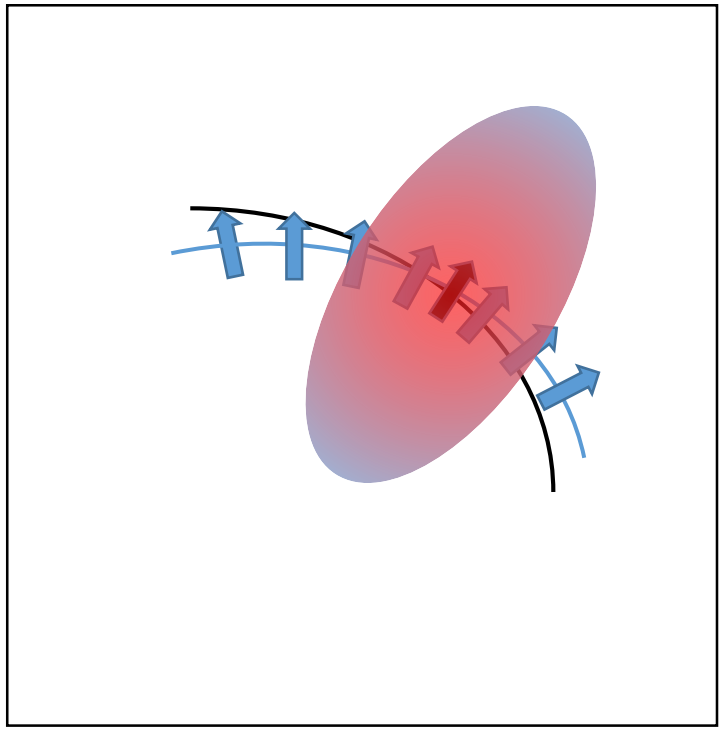
Similarity shape gets aligned with gradient orientation
→ Potentially handling over-counting along edges



Patch $\mathcal{P}$



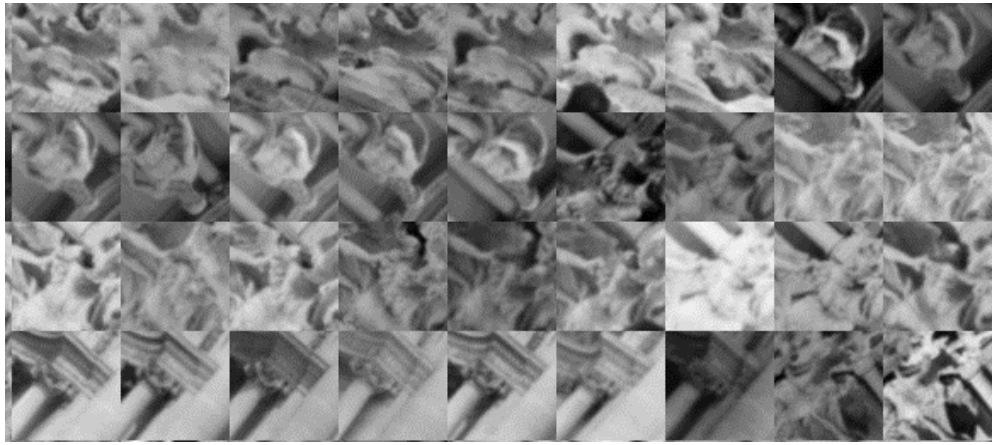
Patch $\mathcal{Q}$



Both: overlaid

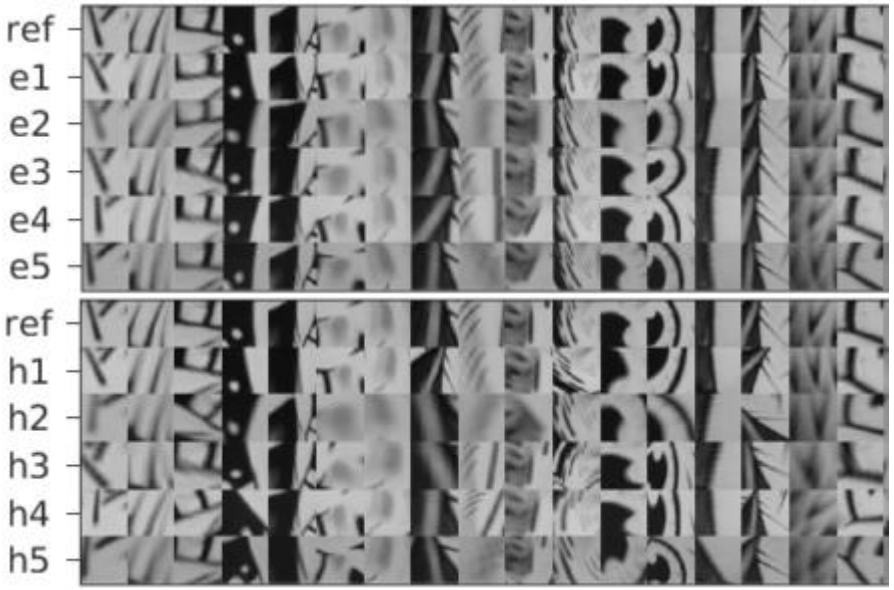
Datasets

Phototourism (PT) – ‘07



[Winder & Brown, CVPR ‘07]

HPatches (HP) – ‘17



[Balntas et al, CVPR ‘17]

Results – PhotoToursim

- Verification task: given 2 patches determine if they depict the same 3D point
- Metric: False positive rate at 95% recall (FPR95)
- Detectors: DoG
- **PCW**: our descriptor (polar + cartes + whitening)

Descriptor	Dimensions	Mean FPR95	CNN	Training time
DC-S2S	512	9.67	×	hours
DDESC	128	9.85	×	hours
Matchnet	4096	7.75	×	hours
TF-M	128	6.47	×	hours
PCW⋆	128	**5.98**		seconds

Results – HPatches

- Tasks: verification, matching, retrieval
- Metric: mAP
- Detectors: DoG, Hessian-Hessian, Harris-Laplace
- PCW: our descriptor (polar + cartes + whitening)

Verification		Matching		Retrieval	
TF–R	81.92	**PCW**	**33.69**	*+TF–R*	40.23
+TF–M	82.69	*+TF–M*	34.29	*+SIFT*	40.36
PCW	**82.94**	*+TF–R*	34.37	*+RSIFT*	43.84
+DC–S2S	83.03	*+DDESC*	35.44	*+DDESC*	44.55
+TF–R	83.24	*+RSIFT*	36.77	**PCW**	**48.26**
PCW⋆	**88.64**	**PCW⋆**	**43.81**	**PCW⋆**	**61.21**

Others: trained on part of Phototourism, whitening learned on part of HPatches

PCW: Whitening learned on part of Phototourism

PCW*: Whitening learned on part of HPatches

Impact of the whitening

Method	Verification	Matching	Retrieval
polar [9]	80.77	32.51	48.04
cartes	70.67	15.79	30.73
polar + *cartes*	77.97	29.34	44.23
polar + *cartes* + *LW*	88.64	43.81	61.21

LW: Learned Whitening

Conclusions

- Hand-crafted descriptor
  - Intuitive design
  - Easy understanding

- Supervised whitening
  - Requires few training pairs
  - Extremely fast to learn
  - Significant performance boost
  - Understanding of the learned similarity

- Outperforms CNN-based descriptors