

0926 Starts Around Page 20  
This is the full slide deck.

# Case Study for Geometric Invariance Local Image Features

EECS 442 Computer Vision

Instructor: Jason Corso (jjcorso)  
[web.eecs.umich.edu/~jjcorso/t/](http://web.eecs.umich.edu/~jjcorso/t/)

# Plan

- What are local image features and why are they useful.
- Local Image Feature Detection
- Invariance
- Local Image Feature Description

# Consider an Application: Image Stitching

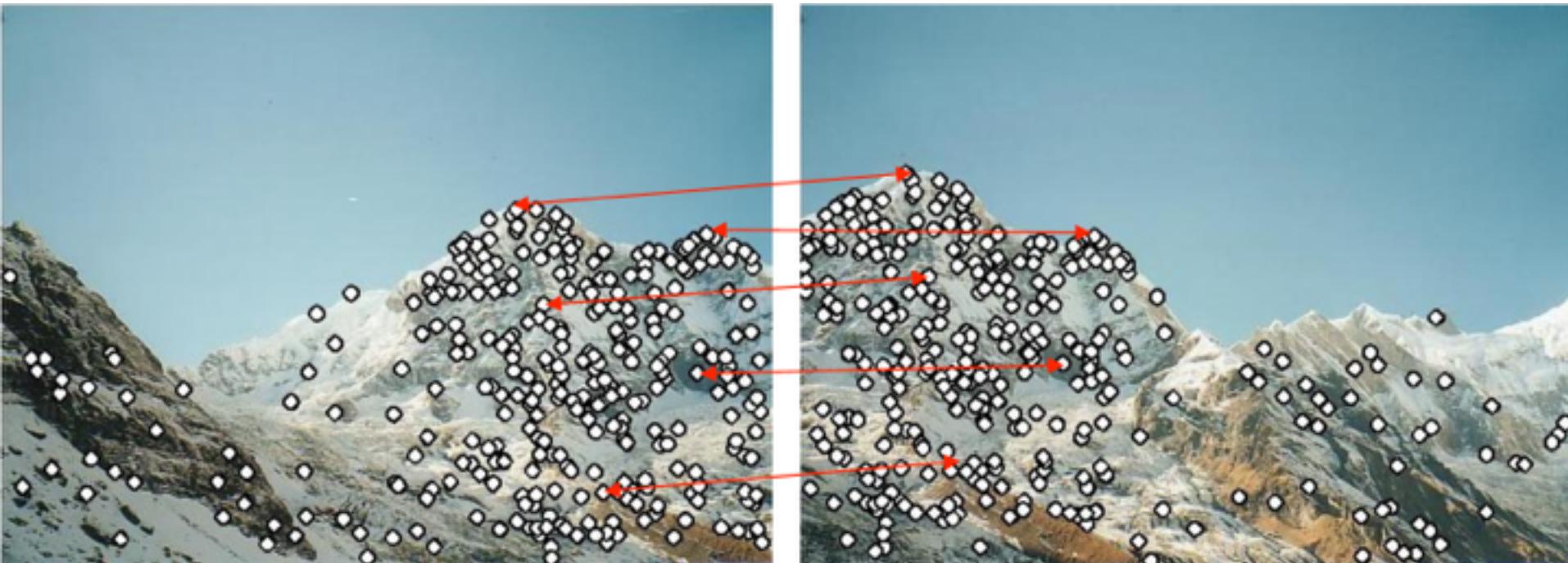


# Consider an Application: Image Stitching



1. Detect feature points in both images.

# Consider an Application: Image Stitching



1. Detect feature points in both images.
2. Find corresponding pairs of feature points.

# Consider an Application: Image Stitching



1. Detect feature points in both images.
2. Find corresponding pairs of feature points.
3. Use the pairs to align the images.

# Consider an Application: Image Stitching



1. Detect feature points in both images. **Reduction**
2. Find corresponding pairs of feature points. **Matching**
3. Use the pairs to align the images. **Estimation**

# What invariants do we care about here?



# What invariants do we care about here?



# What invariants do we care about here?



1. Geometric Invariants
  1. Shift or Translation
  2. Scale? Rotation?
  3. Affine?
  4. Viewpoint?

# What invariants do we care about here?



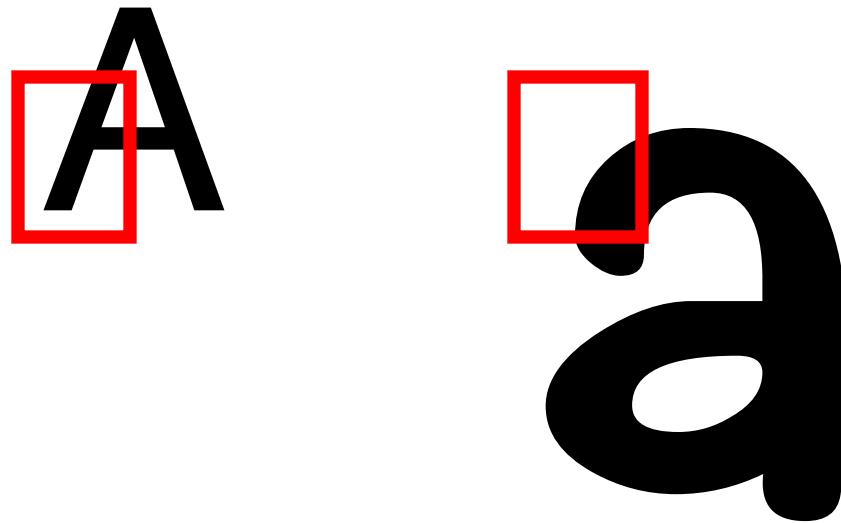
1. Geometric Invariants
  1. Shift or Translation
  2. Scale? Rotation?
  3. Affine?
  4. Viewpoint?
2. Scene layout?

# What invariants do we care about here?



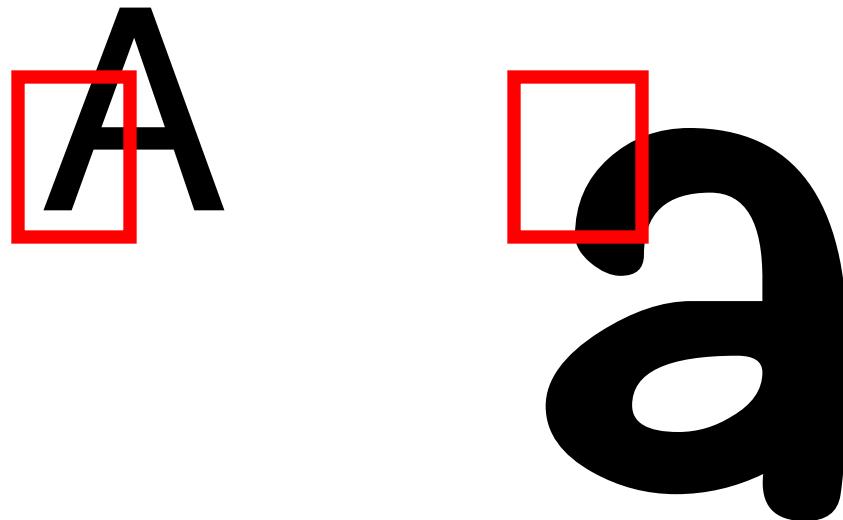
1. Geometric Invariants
  1. Shift or Translation
  2. Scale? Rotation?
  3. Affine?
  4. Viewpoint?
2. Scene layout?
3. Photometric invariants?

# Consider an Application: Detect Object Instances



1. Detect feature points in both images.
2. Find corresponding pairs of feature points.
3. Use the pairs to match object instances.

# What invariants do we care about here?

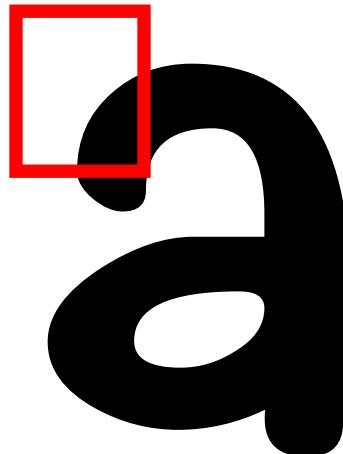
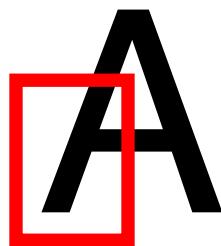


# What invariants do we care about here?

A

a

# What invariants do we care about here?



1. Geometric Invariants
  1. Shift or Translation
  2. Scale? Rotation?
  3. Affine?
  4. Viewpoint?
2. Scene layout?
3. Photometric invariants?
4. Character shape invariance? “Font” invariance.

# Case Study in Local Image Features

- Basic flow of applications in the case study
  1. Detect feature points in both images.
  2. Find corresponding pairs of feature points.
  3. Use the pairs to solve objective function.

# Case Study in Local Image Features

- Basic flow of applications in the case study
  1. Detect feature points in both images.
  2. Find corresponding pairs of feature points.
  3. Use the pairs to solve objective function.

**Reduction  
Matching  
Estimation**

# Case Study in Local Image Features

- Basic flow of applications in the case study
  1. Detect feature points in both images.
  2. Find corresponding pairs of feature points.
  3. Use the pairs to solve objective function.
- Other applications of local image features
  - 3D reconstruction
  - Motion tracking
  - Object recognition
  - Indexing and database retrieval
  - Robot navigation

**Reduction  
Matching  
Estimation**

# Advantages of local features

## Locality

- features are local, so robust to occlusion and clutter

## Distinctiveness:

- can differentiate a large database of objects

## Quantity

- hundreds or thousands in a single image

## Efficiency

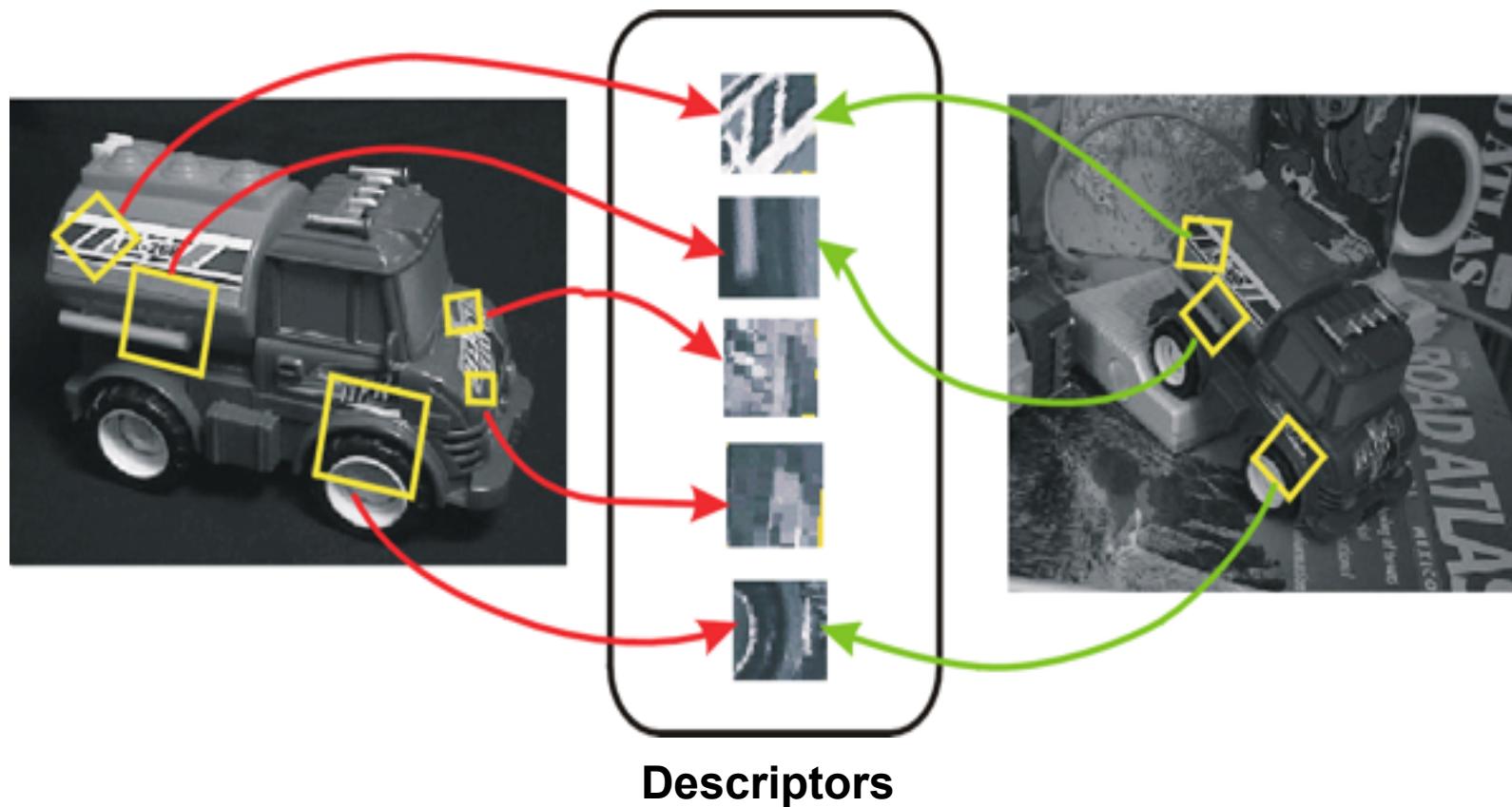
- real-time performance achievable

## Generality

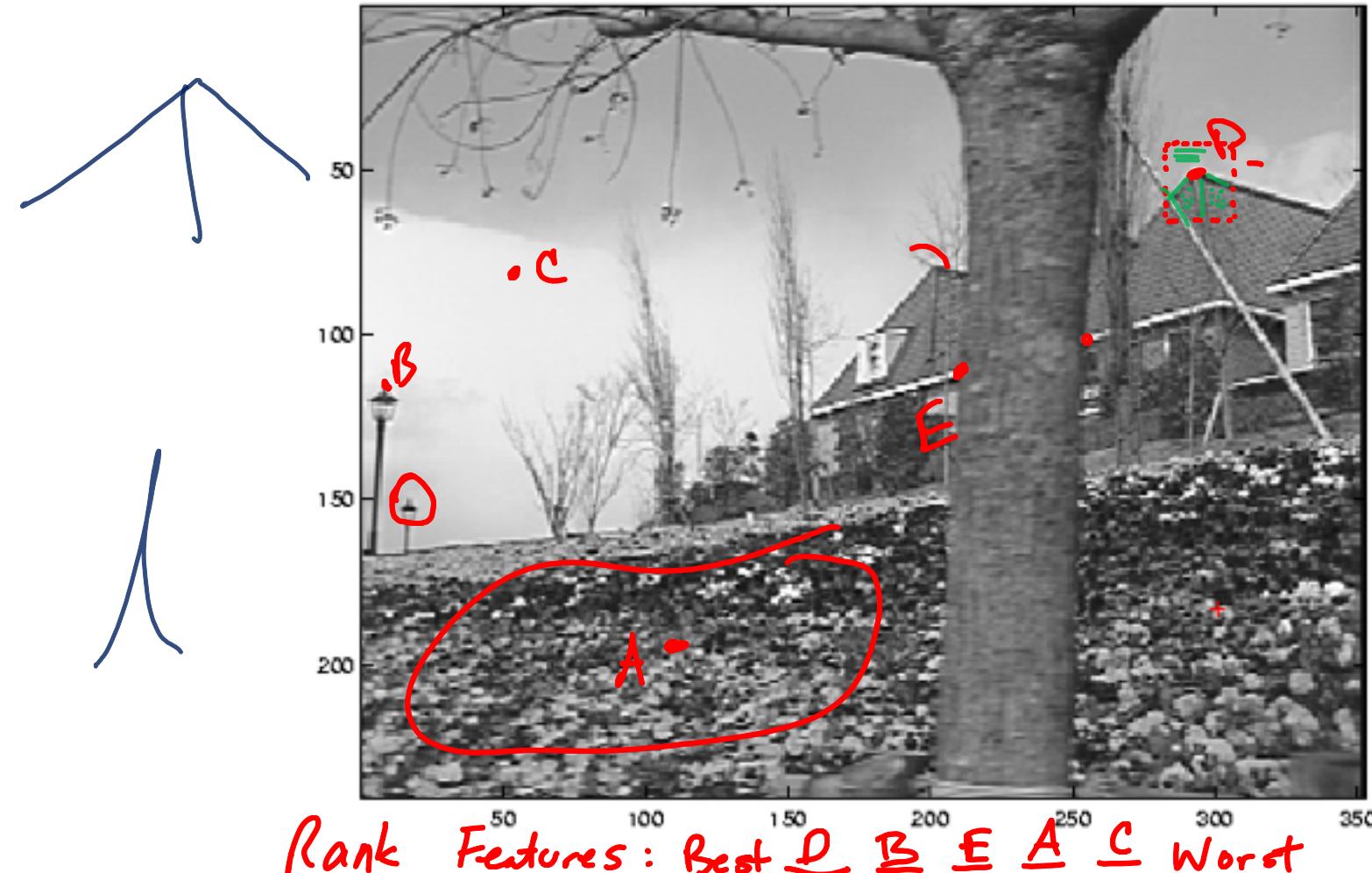
- exploit different types of features in different situations

# Challenges

- Repeatability
- Uniqueness
- Invariance w.r.t. Matching



# What makes a good feature?



# Repeatability



Illumination  
invariance

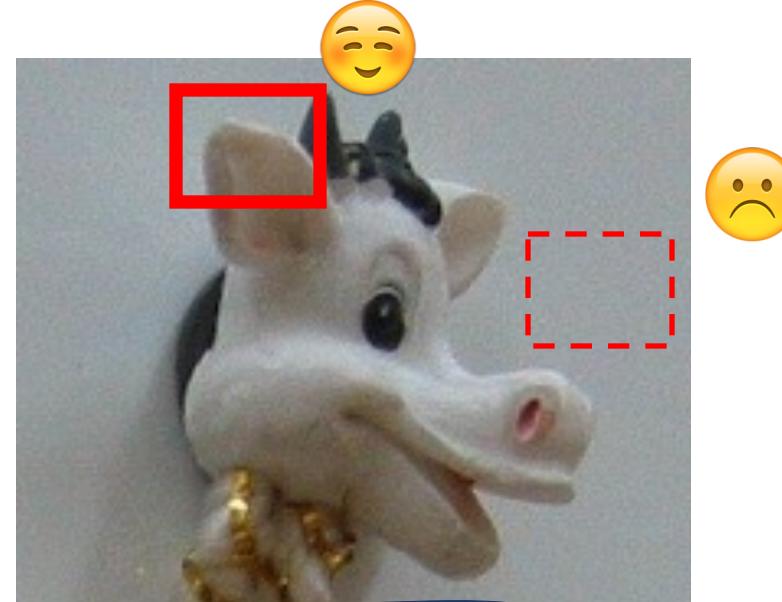


Scale  
invariance

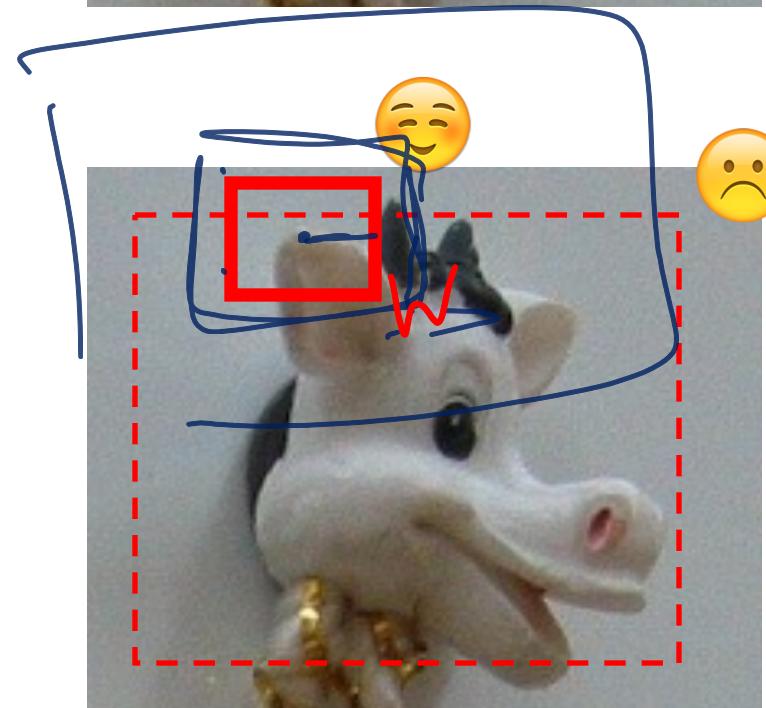


Pose invariance  
• Rotation   
• Affine

- Saliency



- Locality



# One criterion is uniqueness

Look for image regions that are unusual

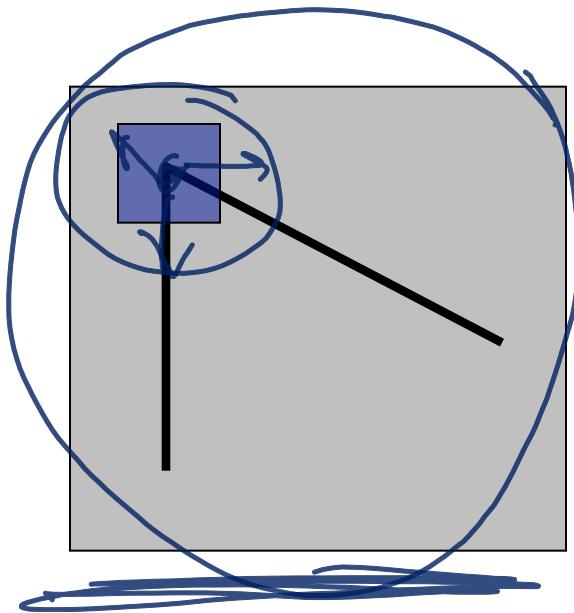
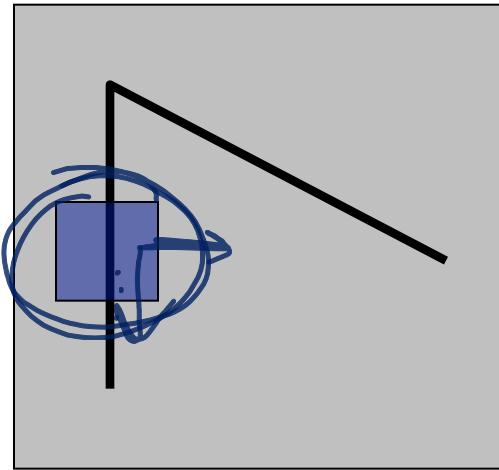
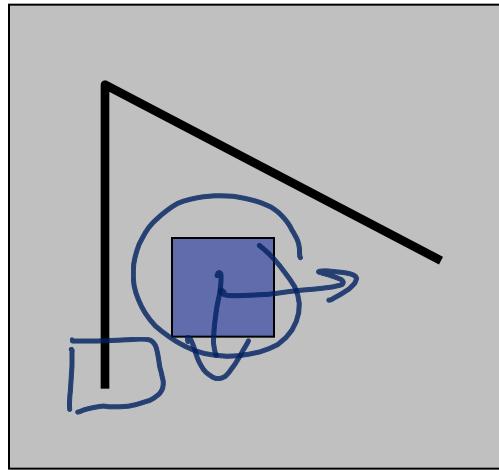
- Lead to unambiguous matches in other images
- 

How to define “unusual”?

# Local measures of uniqueness

Suppose we only consider a small window of pixels

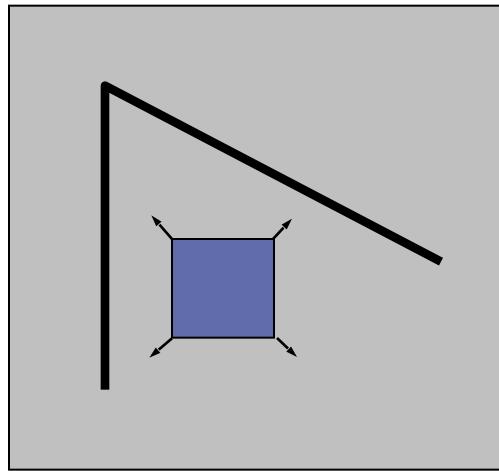
- What defines whether a feature is a good or bad candidate?



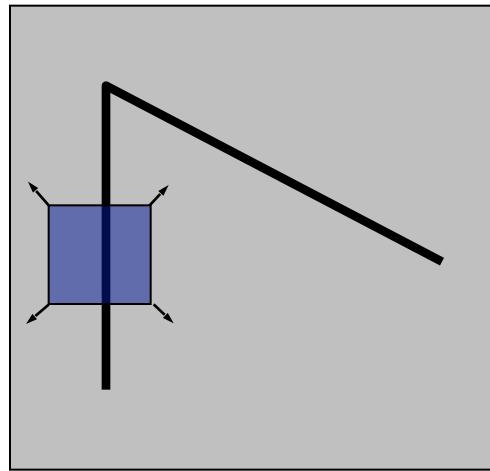
# Feature detection

Local measure of feature uniqueness

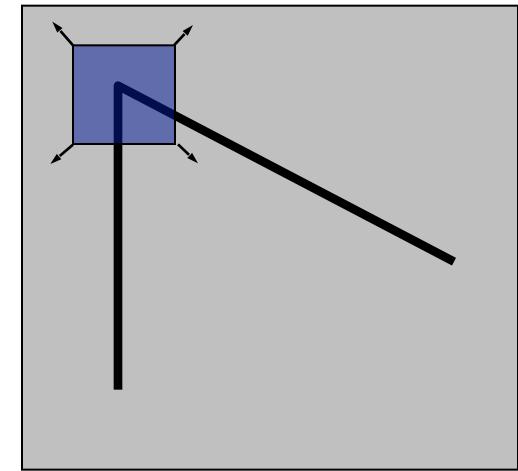
- How does the window change when you shift it?
- Shifting the window in *any direction* causes a *big change*



“flat” region:  
no change in all  
directions



“edge”:  
no change along the  
edge direction



“corner”:  
significant change in  
all directions

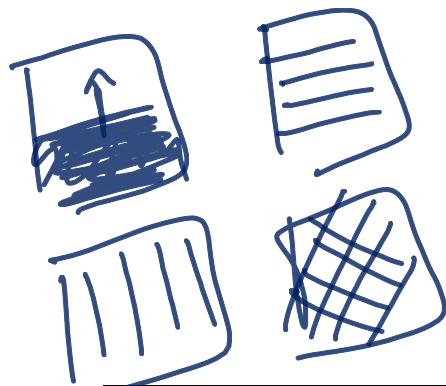
# Stop Slides

See hand-written lecture notes for the mathematical derivation of the corner operator.

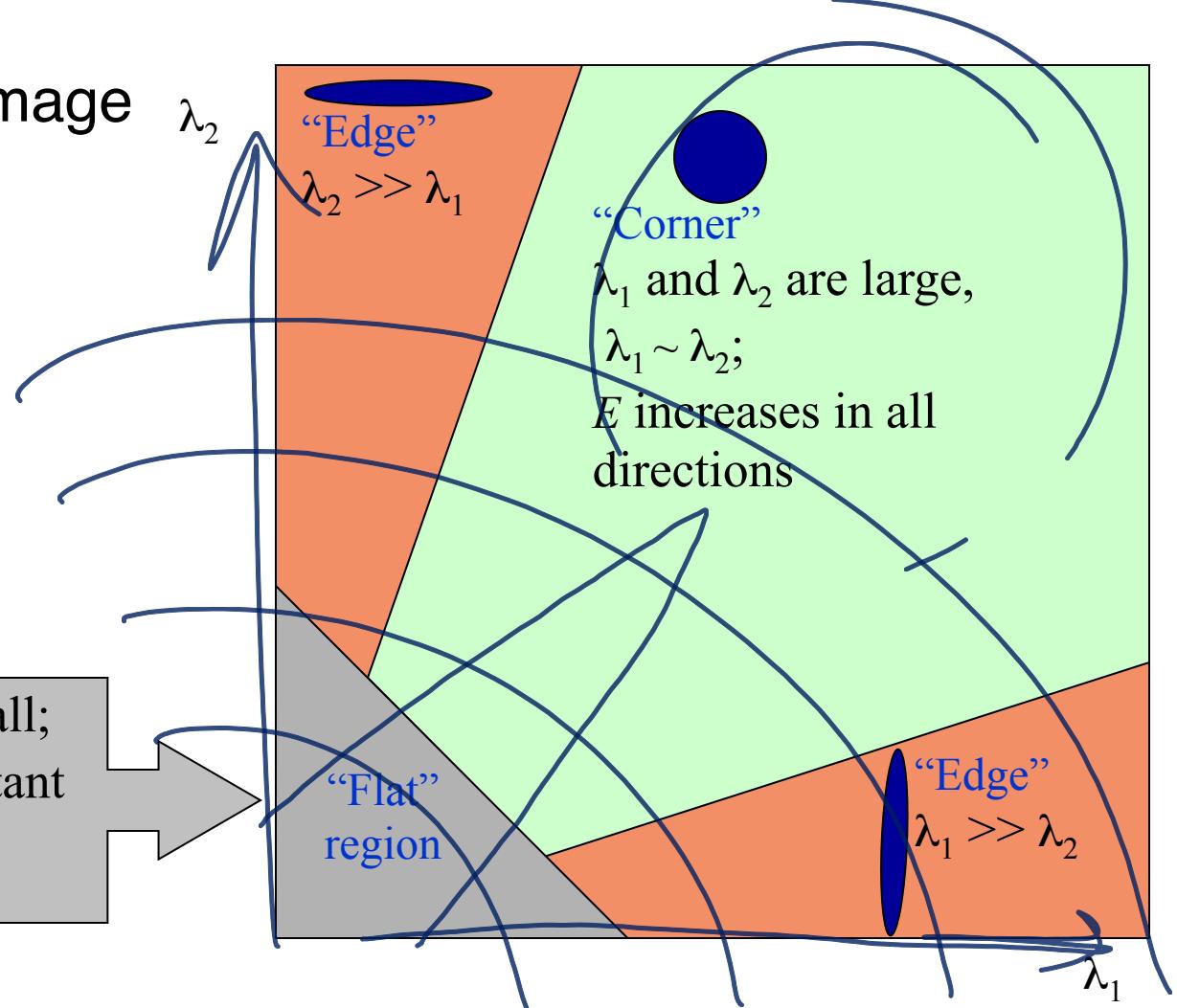
# Feature detection: the math

$\lambda_1, \lambda_2$  are eigenvalues of structr tensor

Classification of image points using eigenvalues of  $M$ :



$\lambda_1$  and  $\lambda_2$  are small;  
 $E$  is almost constant  
in all directions



# Feature detection: the math



$$H(I, x, y, w_s)$$

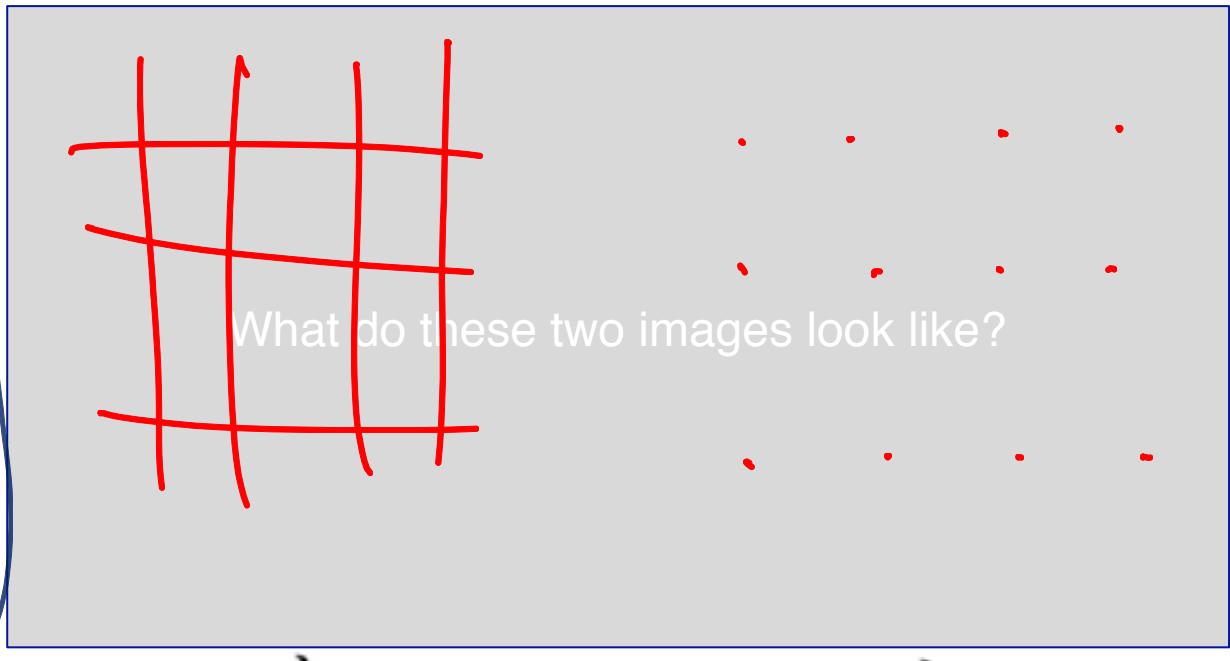
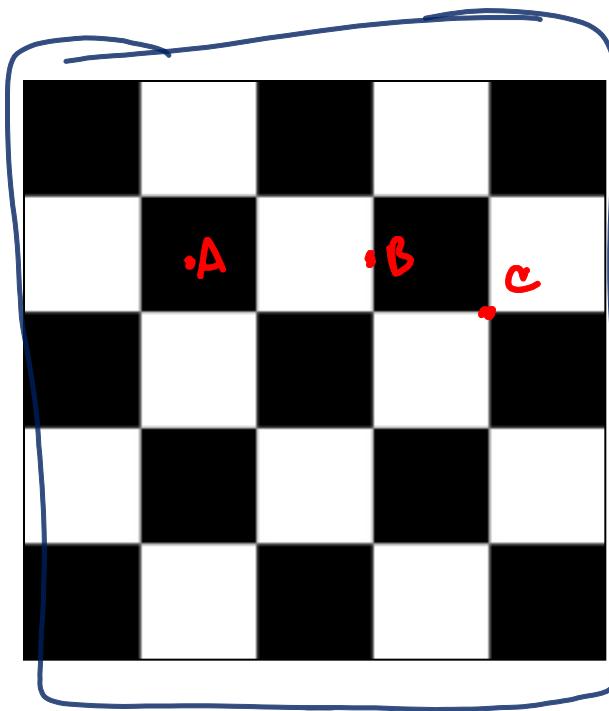
How are  $\lambda_+$ ,  $x_+$ ,  $\lambda_-$ , and  $x_-$  relevant for feature detection?

- What's our feature scoring function?

$$\lambda_+(x, y) = \lambda_+(H(I, x, y, w_s))$$

Want  $E(T)$  to be *large* for small shifts in *all* directions

- the *minimum* of  $E(T)$  should be large, over all unit vectors  $[u v]$
- this minimum is given by the smaller eigenvalue ( $\lambda_-$ ) of  $H$



$I$

$\lambda_+$

$\lambda_-$

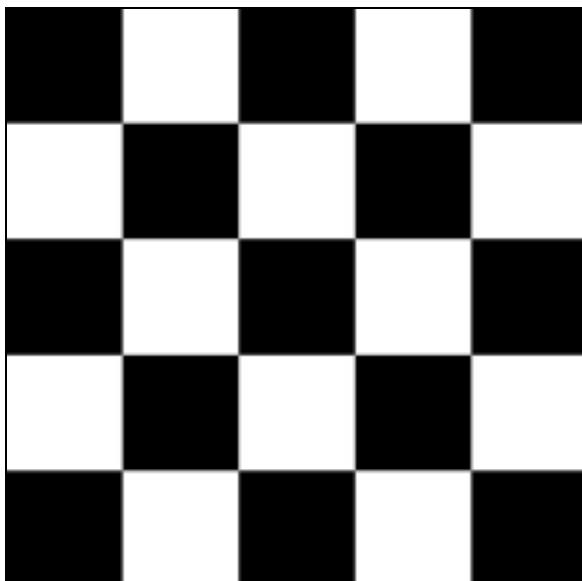
# Feature detection: the math

How are  $\lambda_+$ ,  $x_+$ ,  $\lambda_-$ , and  $x_-$  relevant for feature detection?

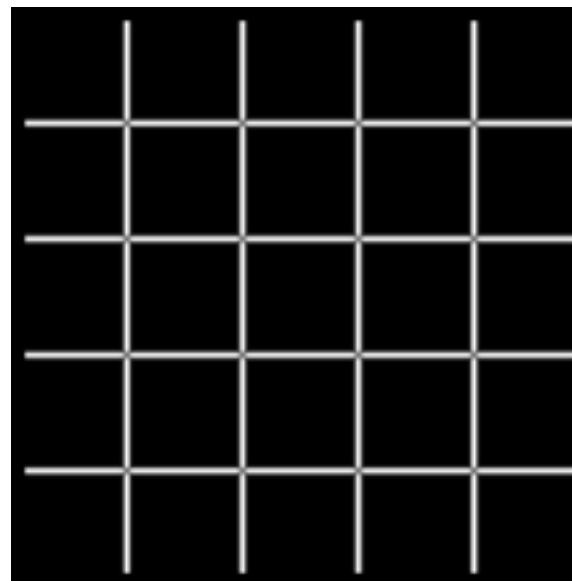
- What's our feature scoring function?

Want  $E(T)$  to be *large* for small shifts in *all* directions

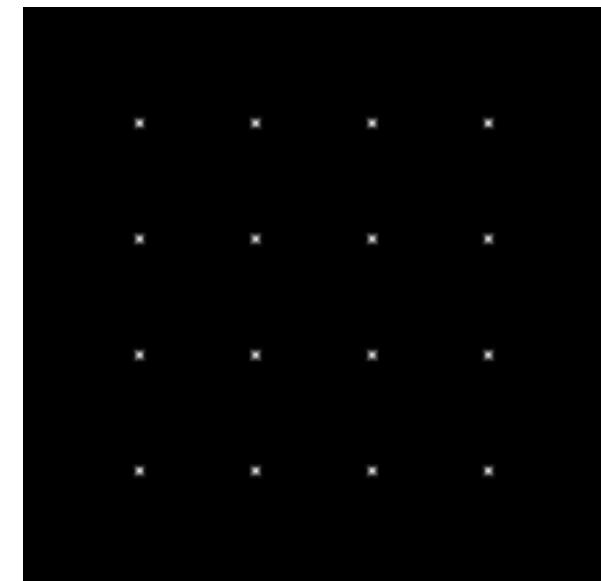
- the *minimum* of  $E(T)$  should be large, over all unit vectors  $[u \ v]$
- this minimum is given by the smaller eigenvalue ( $\lambda_-$ ) of  $H$



$I$



$\lambda_+$

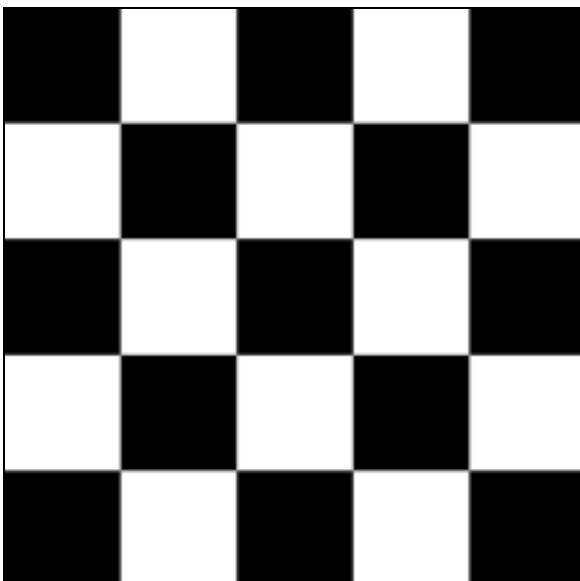
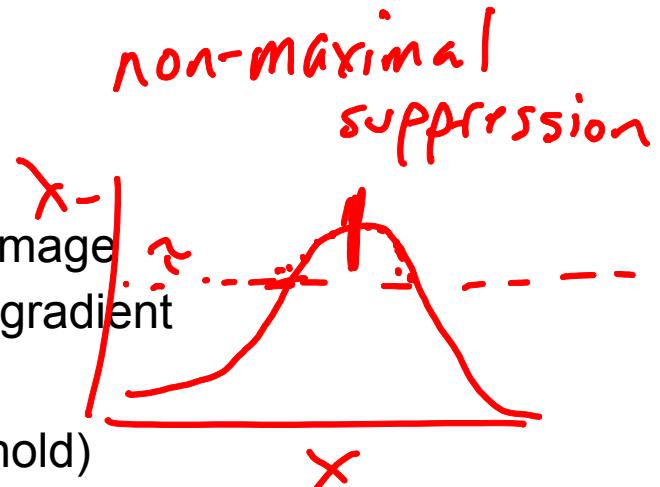


$\lambda_-$

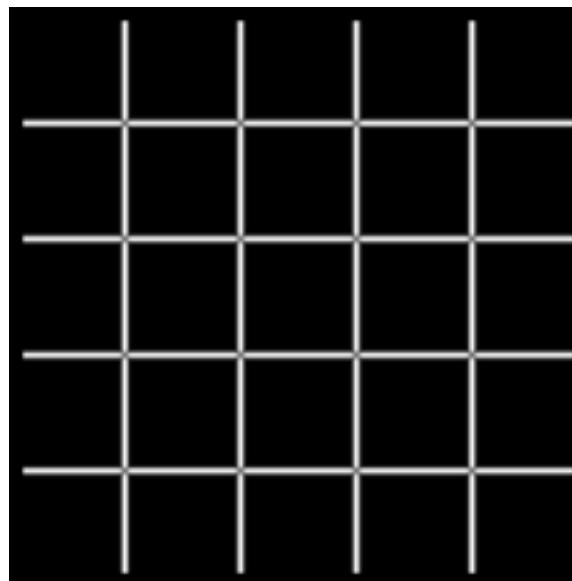
# Feature detection summary

Here's what you do

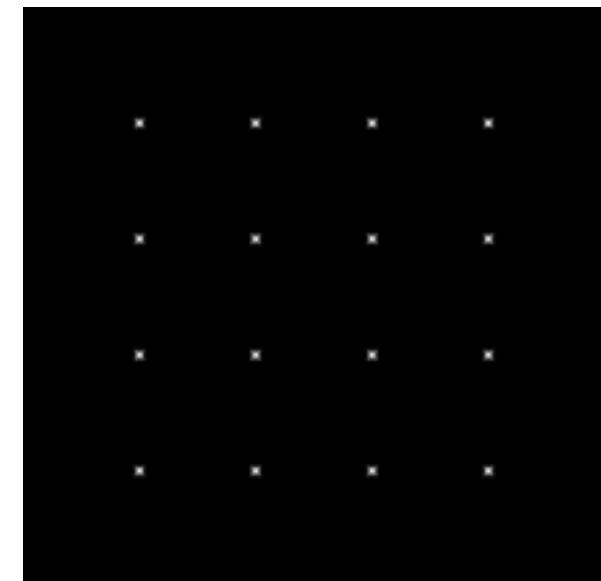
- Compute the gradient at each point in the image
- Create the  $H$  matrix from the entries in the gradient
- Compute the eigenvalues.
- Find points with large response ( $\lambda_- > \text{threshold}$ )
- Choose those points where  $\lambda_-$  is a local maximum as features



$I$



$\lambda_+$

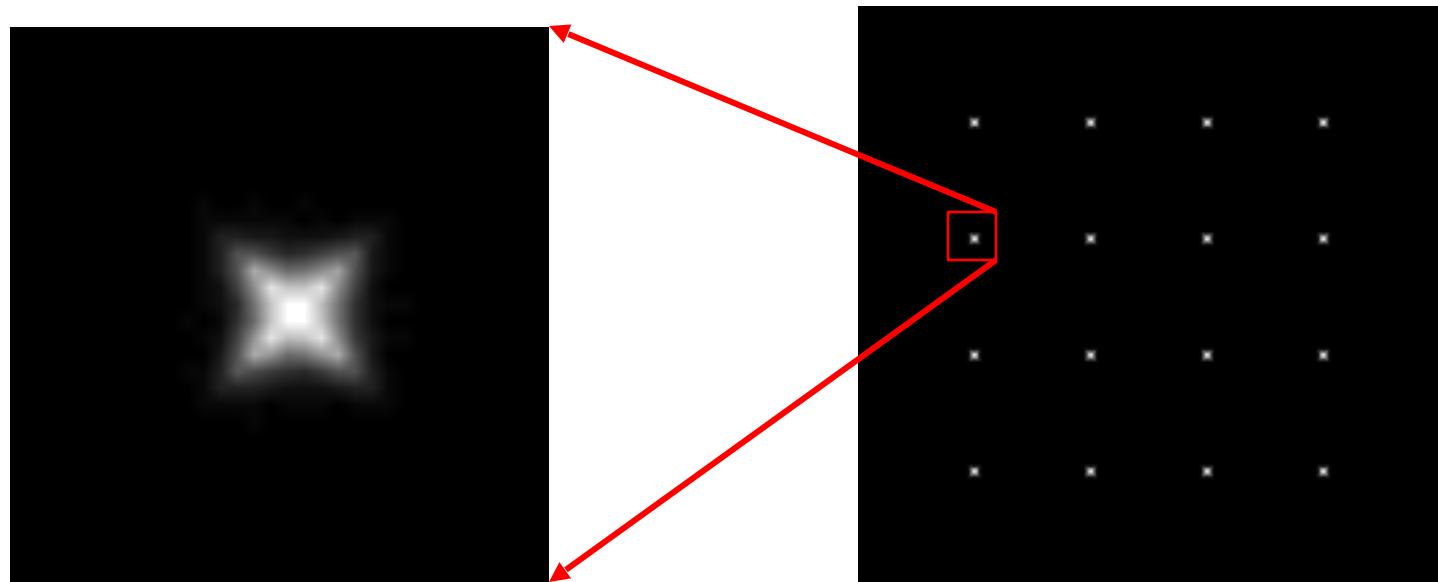


$\lambda_-$

# Feature detection summary

Here's what you do

- Compute the gradient at each point in the image
- Create the  $H$  matrix from the entries in the gradient
- Compute the eigenvalues.
- Find points with large response ( $\lambda_- > \text{threshold}$ )
- Choose those points where  $\lambda_-$  is a local maximum as features



$\lambda_-$

# The Harris operator

$\lambda_{\_}$  is a variant of the “Harris operator” for feature detection

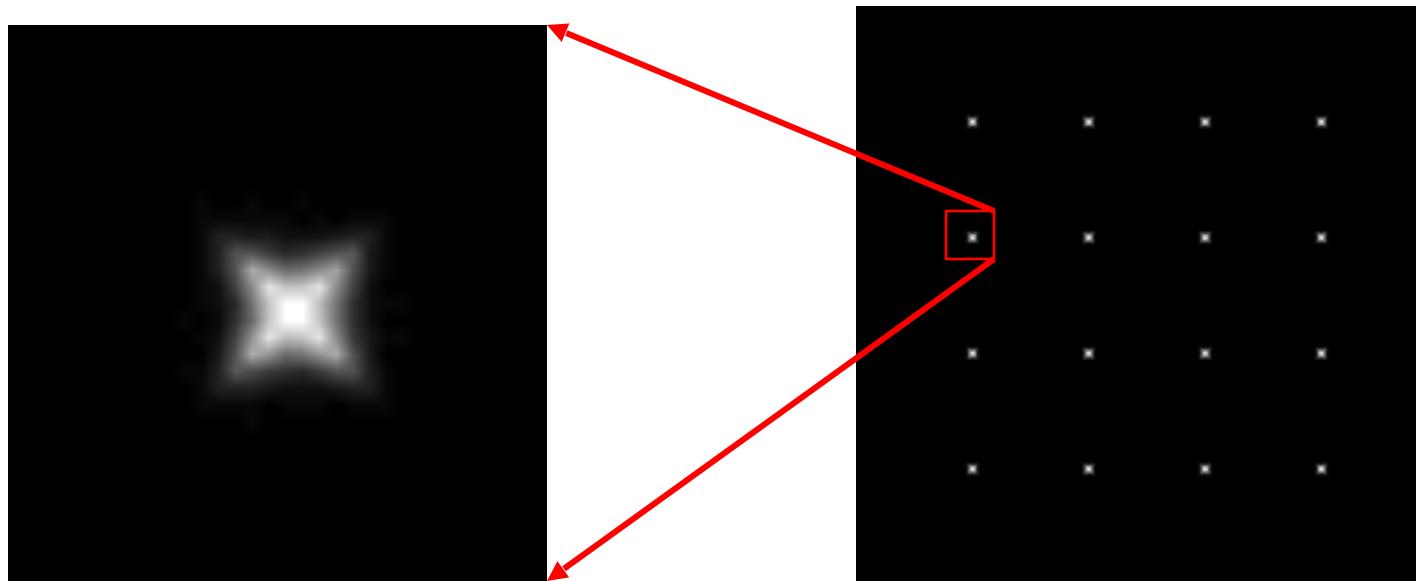
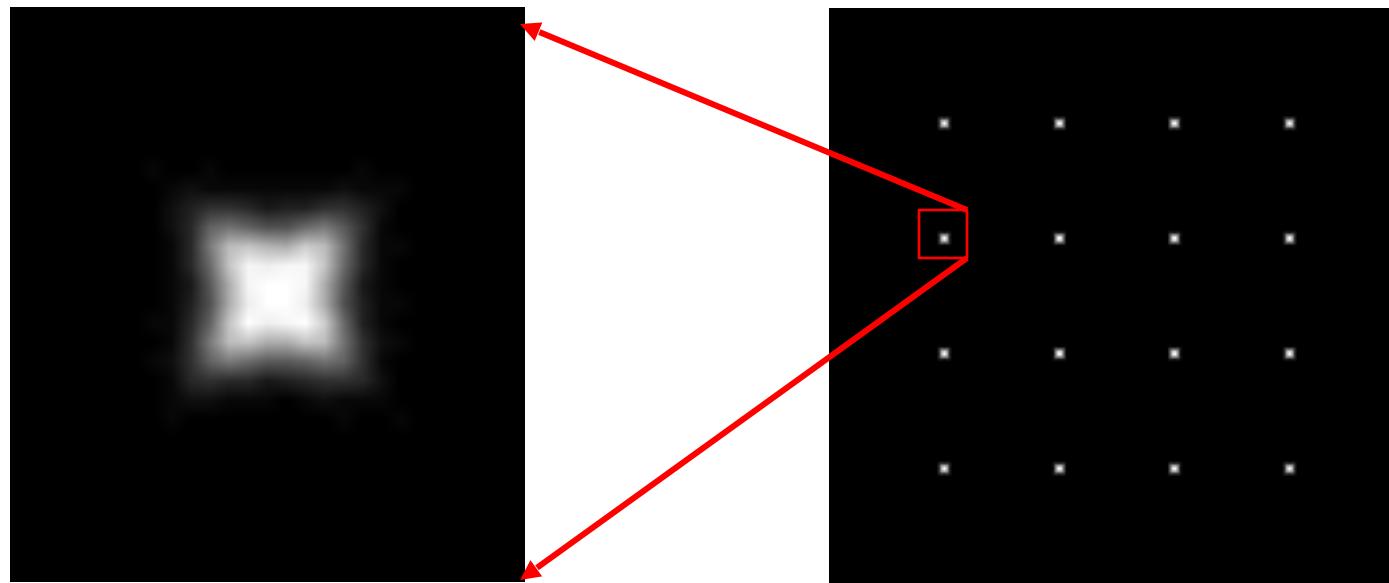
$$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$

$$= \frac{\text{determinant}(H)}{\text{trace}(H)}$$

- The *trace* is the sum of the diagonals, i.e.,  $\text{trace}(H) = h_{11} + h_{22}$
- Very similar to  $\lambda_{\_}$  but less expensive (no square root)
- Called the “Harris Corner Detector” or “Harris Operator”
- Lots of other detectors, this is one of the most popular

→ C.Harris and M.Stephens. "A Combined Corner and Edge Detector."  
*Proceedings of the 4th Alvey Vision Conference:* pages 147–151. 1988.

# The Harris operator

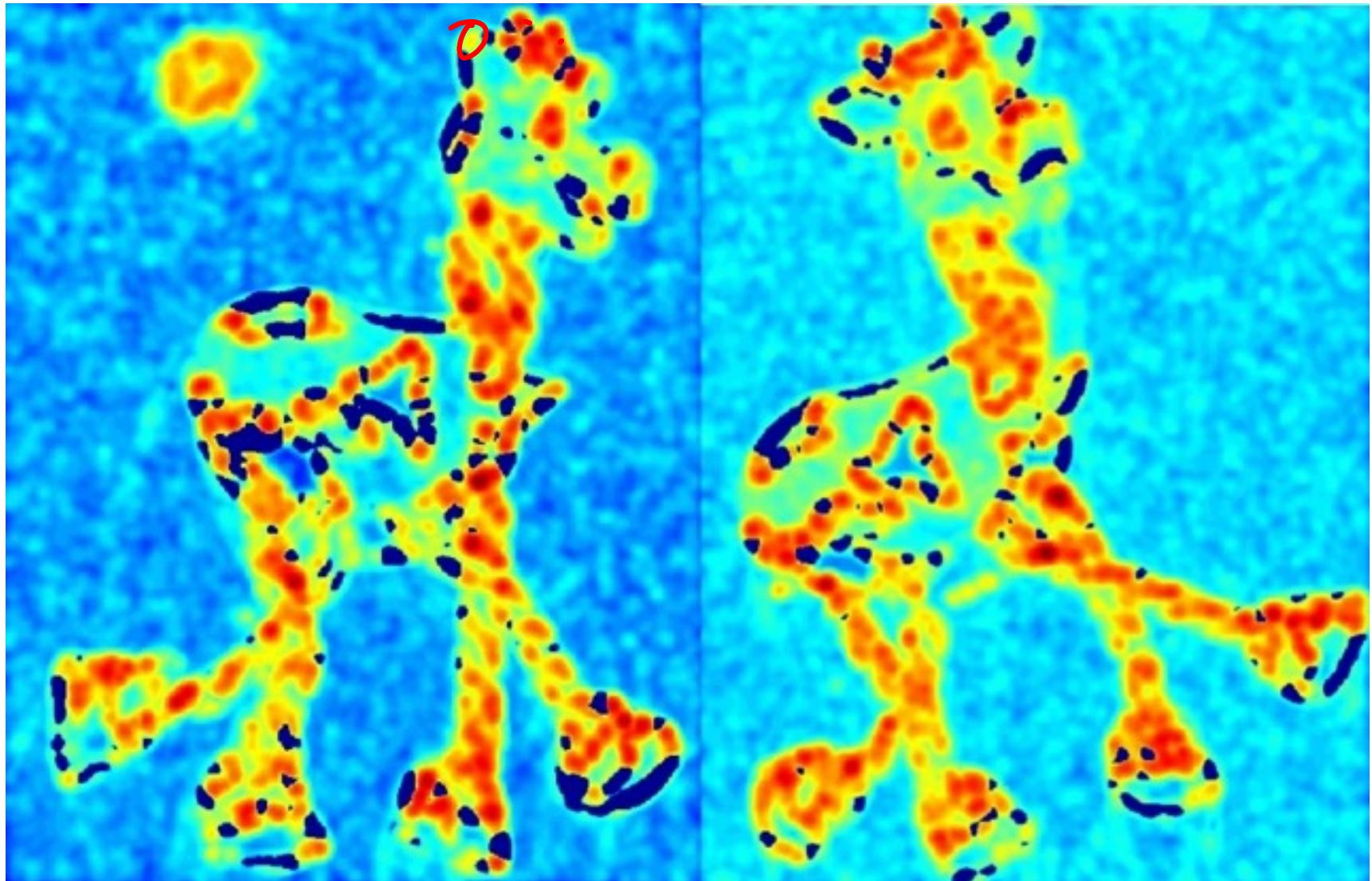


# Harris detector example

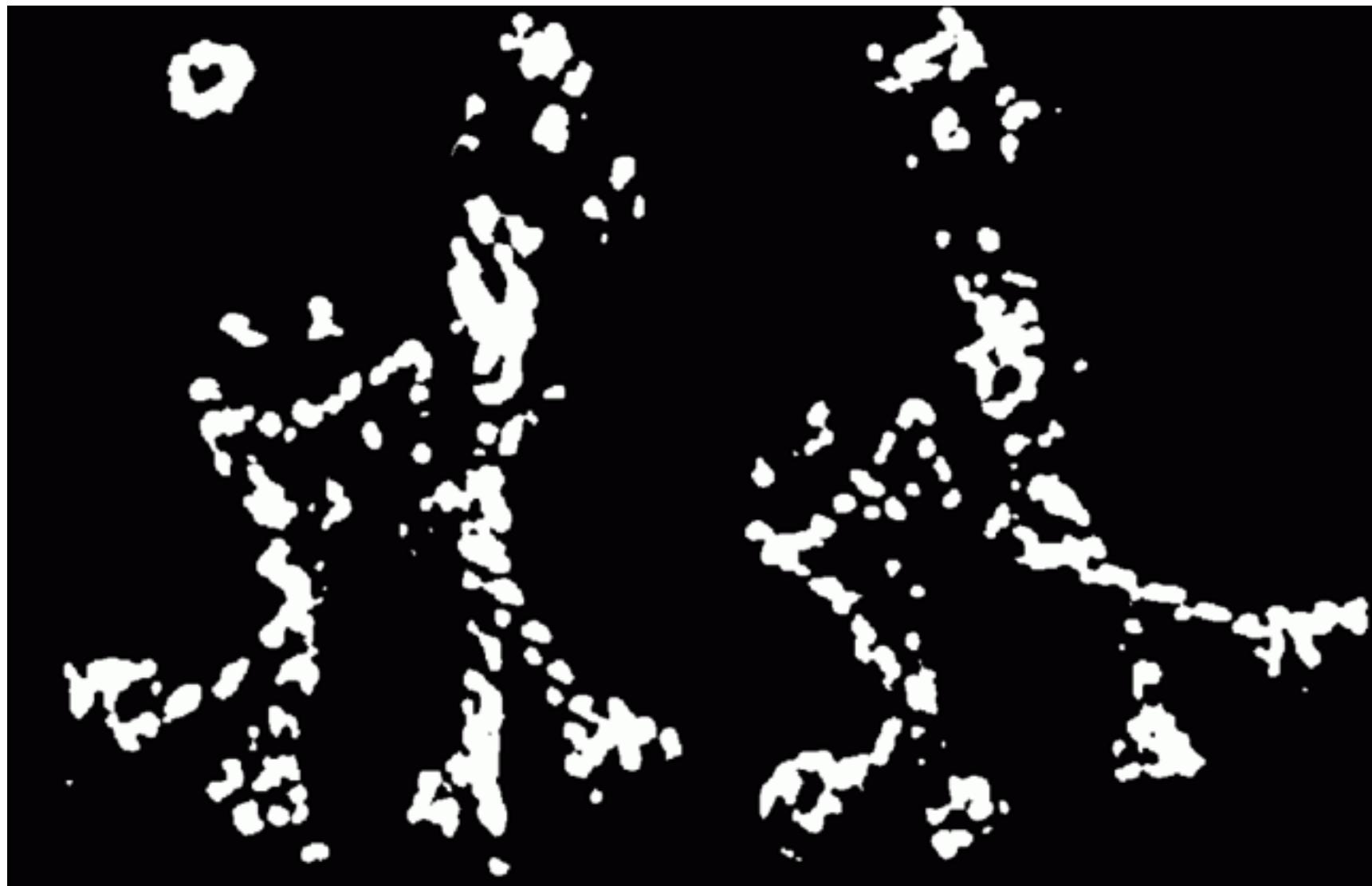


f value (red high, blue low)

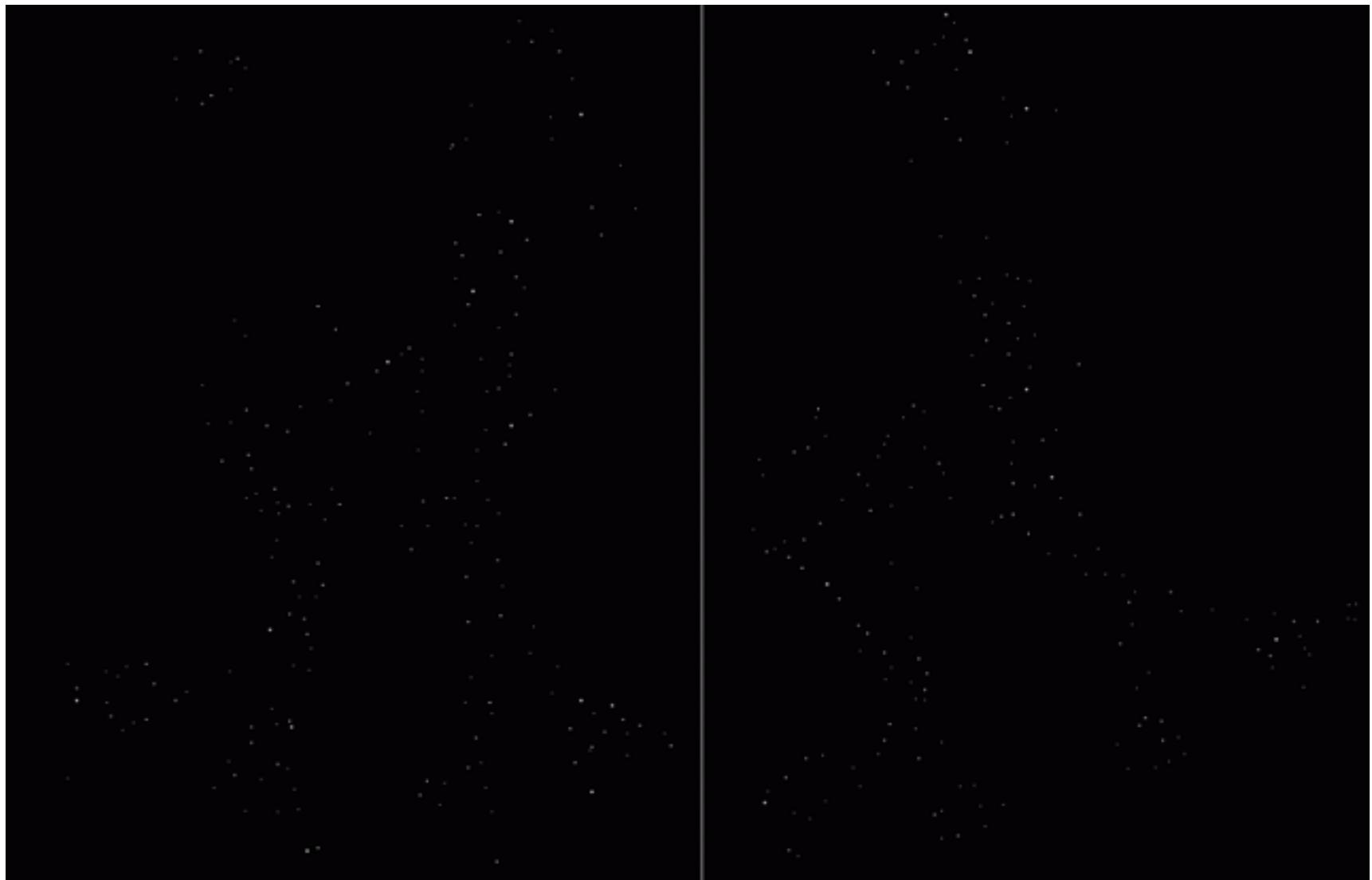
$$\frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$



## Threshold ( $f > \text{value}$ )



# Find local maxima of $f$



## Harris features (in red)



# Stop Slides

End of Corner Detector Lecture  
(Invariance of Corner Detector Follows in  
Hand-written notes)

# Towards Invariance

Suppose you **rotate** the image by some angle

- Will you still pick up the same features?

What if you change the brightness?

Scale?

# Towards Invariance

Suppose you **rotate** the image by some angle

- Will you still pick up the same features?

What if you change the brightness?

Scale?

**Invariance** defined:

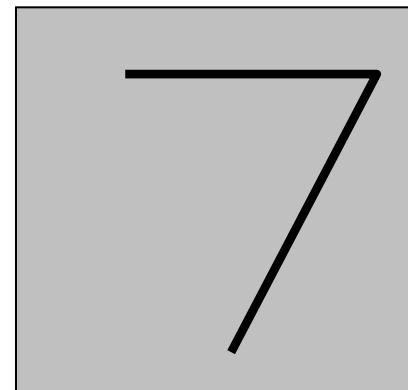
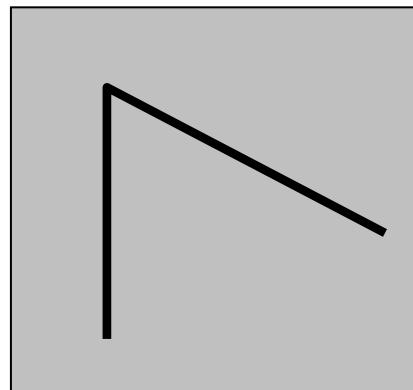
Suppose we are comparing two images I and J.

J may be a transformed version of I

We want to detect the same features from I and J regardless of the transformation: this is **transformational invariance**.

# Harris Detector: Some Properties

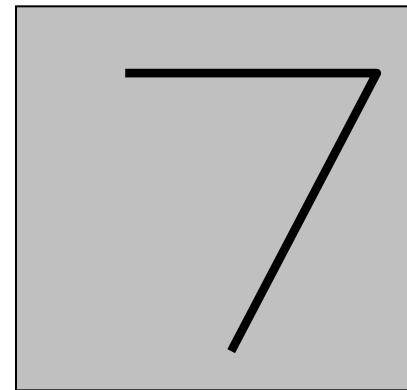
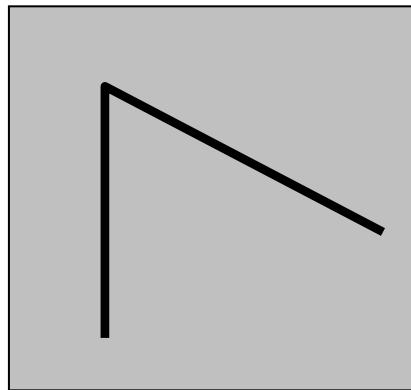
- Is the Harris detector rotationally invariant?



$$H = U^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} U \rightarrow f(\lambda_1, \lambda_2)$$

# Harris Detector: Some Properties

- Is the Harris detector rotationally invariant?



*Corner response  $R$  is invariant to image rotation*

$$H = U^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} U \rightarrow f(\lambda_1, \lambda_2) \text{ doesn't change!}$$

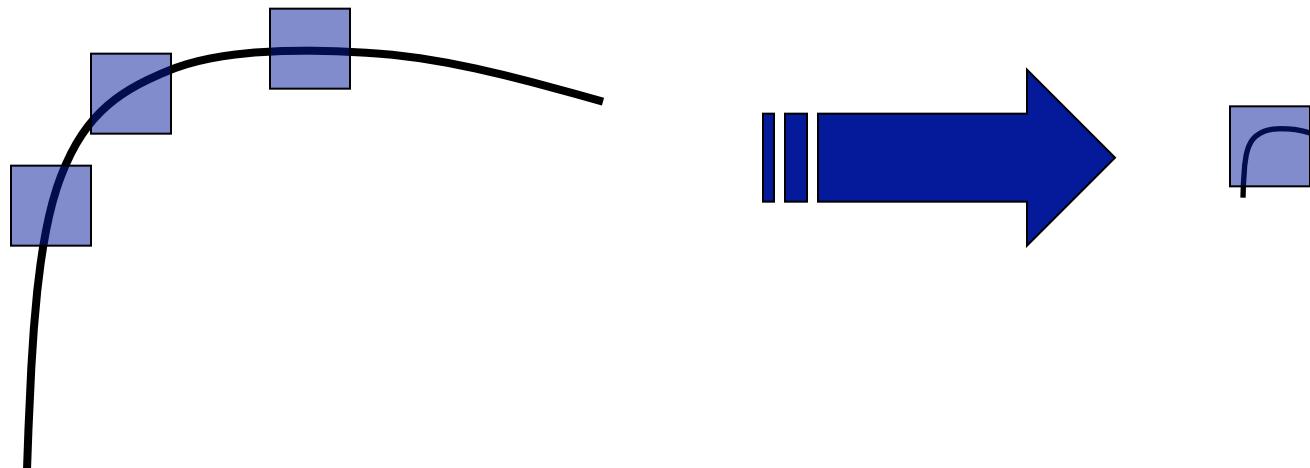
# Harris Detector: Some Properties

- Is it scale invariant?

# Harris Detector: Some Properties

- Is it scale invariant?

*Corner response  $R$  is not scale invariant!*



All points will be  
classified as **edges**

Corner !

# Harris Detector: Some Properties

- Partial invariance to *affine intensity* changes

$$I \rightarrow s I + b$$

# Harris Detector: Some Properties

- Partial invariance to *affine intensity* changes  
 $I \rightarrow s I + b$
- invariance to intensity shift  $I \rightarrow I + b$  (*why?*)

# Harris Detector: Some Properties

- Partial invariance to *affine intensity* changes

$$I \rightarrow s I + b$$

- invariance to intensity shift  $I \rightarrow I + b$  (*why?*)

(only derivatives are used)

# Harris Detector: Some Properties

- Partial invariance to *affine intensity* changes

$$I \rightarrow s I + b$$

- invariance to intensity shift  $I \rightarrow I + b$  (*why?*)

(only derivatives are used)

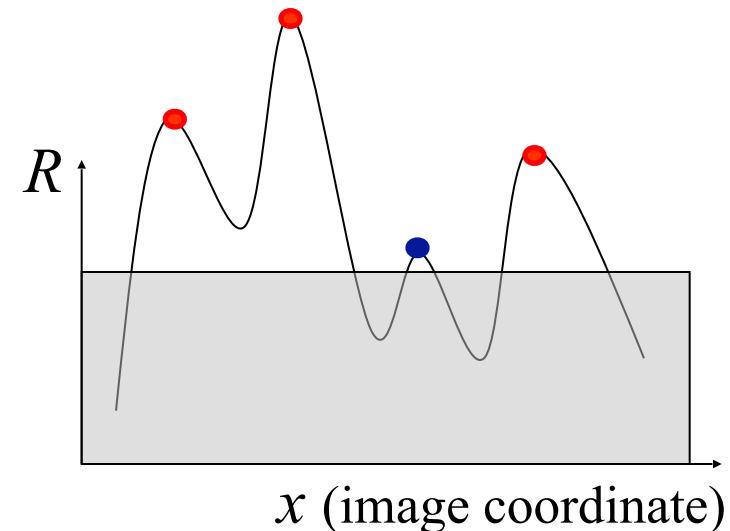
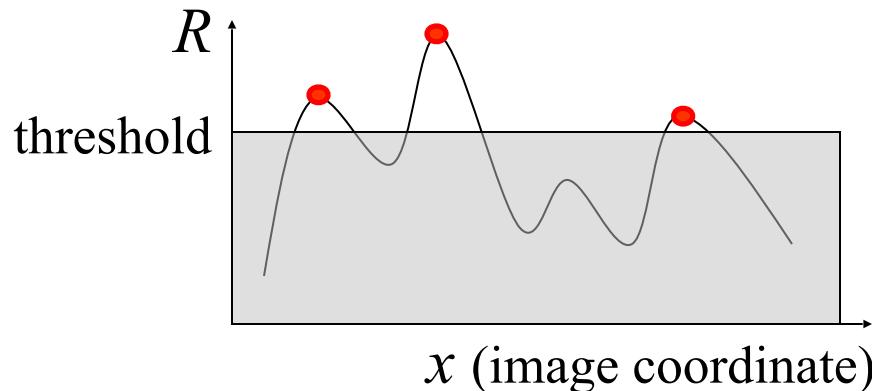
- Not invariant to intensity scale:  $I \rightarrow a I$

# Harris Detector: Some Properties

- Partial invariance to *affine intensity changes*

$$I \rightarrow s I + b$$

- invariance to intensity shift  $I \rightarrow I + b$  (*why?*)  
(only derivatives are used)
- Not invariant to intensity scale:  $I \rightarrow a I$



# Invariance

Detector	Illumination	Rotation	Scale	View point
Harris corner	partial	Yes	No	No

# Next Steps (After Initial Discussion on Description)

- Exploring further invariance in feature detection
  - Scale invariance, scale-space and adaptive scale selection
  - Affine invariance
- Exploring further invariance in feature description
  - Photometric invariance
  - Rotation invariance (noted in this lecture)
  - Affine invariance

# Local Feature Descriptors

# Application: Image Stitching

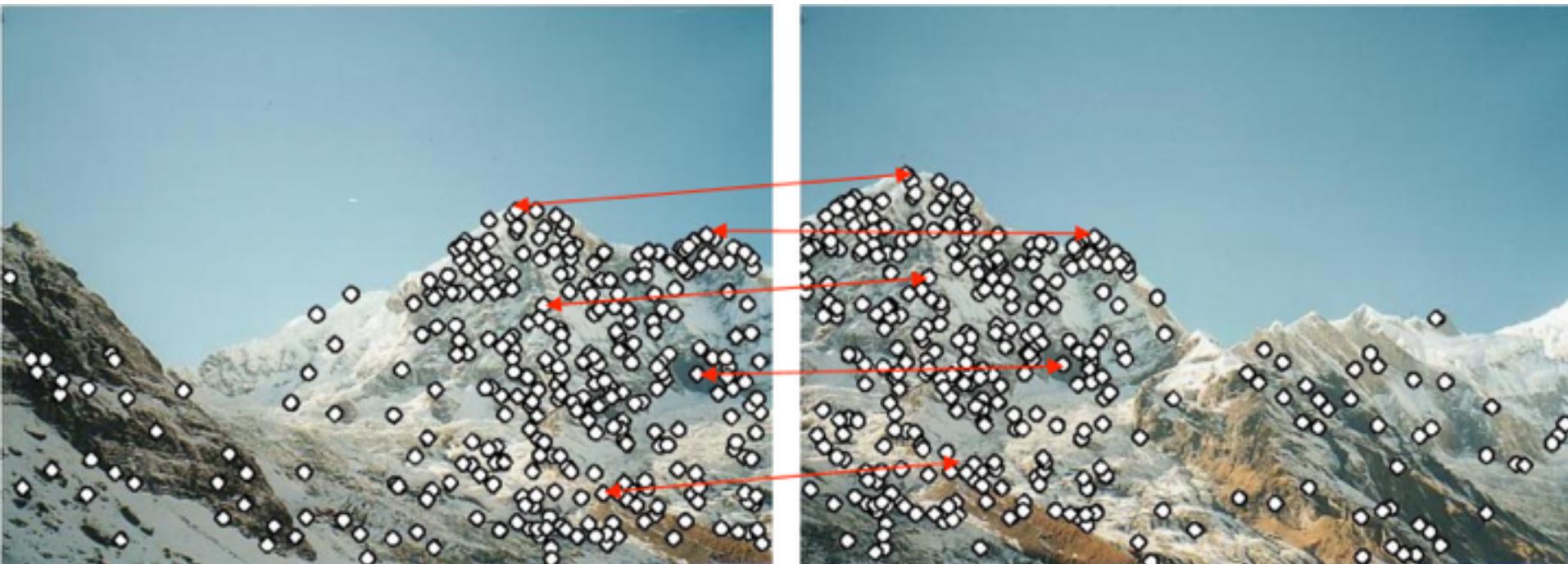


# Application: Image Stitching



1. Detect feature points in both images.

# Application: Image Stitching



1. Detect feature points in both images.
2. Find corresponding pairs of feature points.

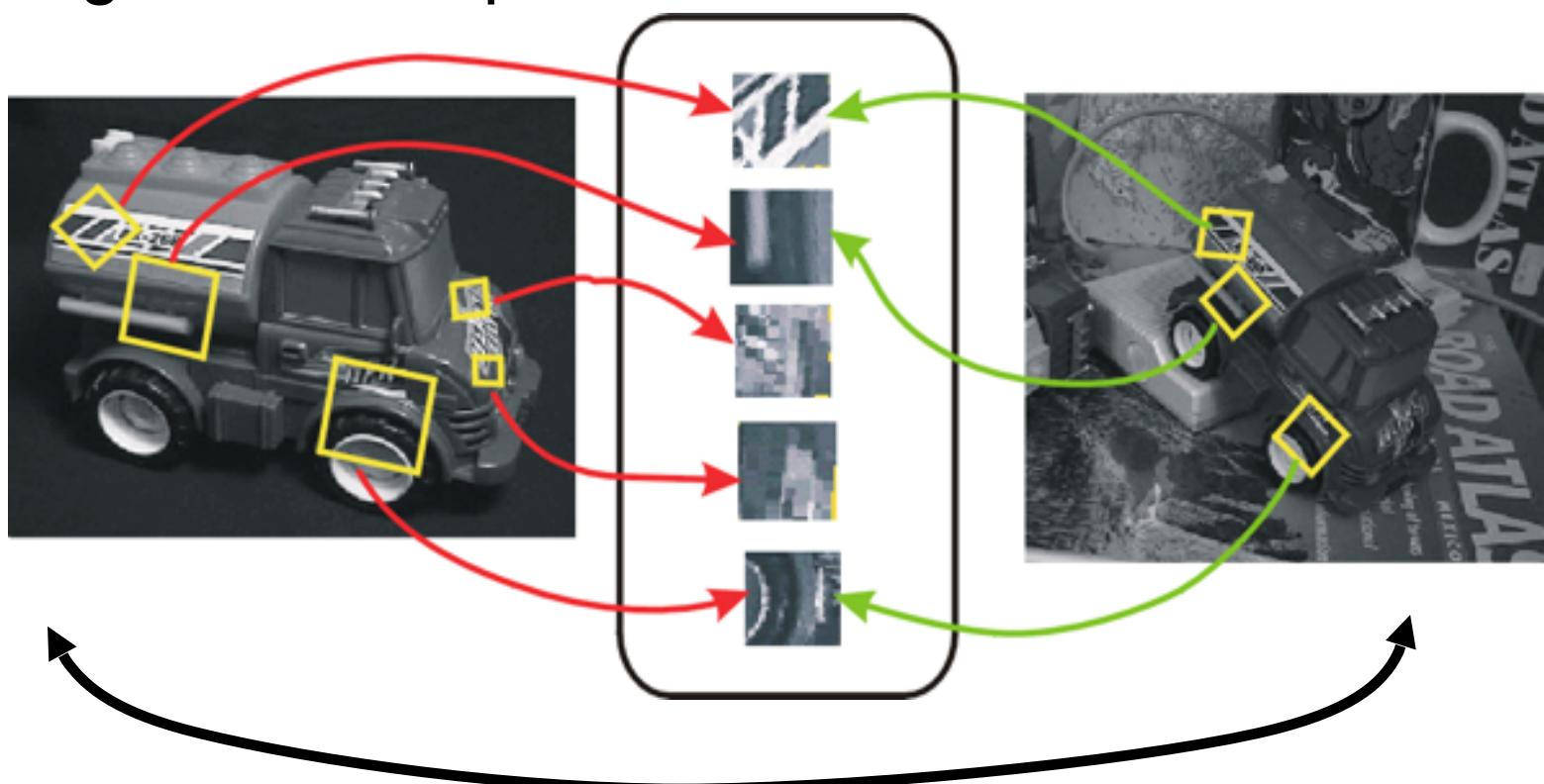
# Application: Image Stitching



1. Detect feature points in both images.
2. Find corresponding pairs of feature points.
3. Use the pairs to align the images.

# Pose normalization

- Keypoints are transformed in order to be invariant to translation, rotation, scale, and other geometrical parameters [Lowe 2000]



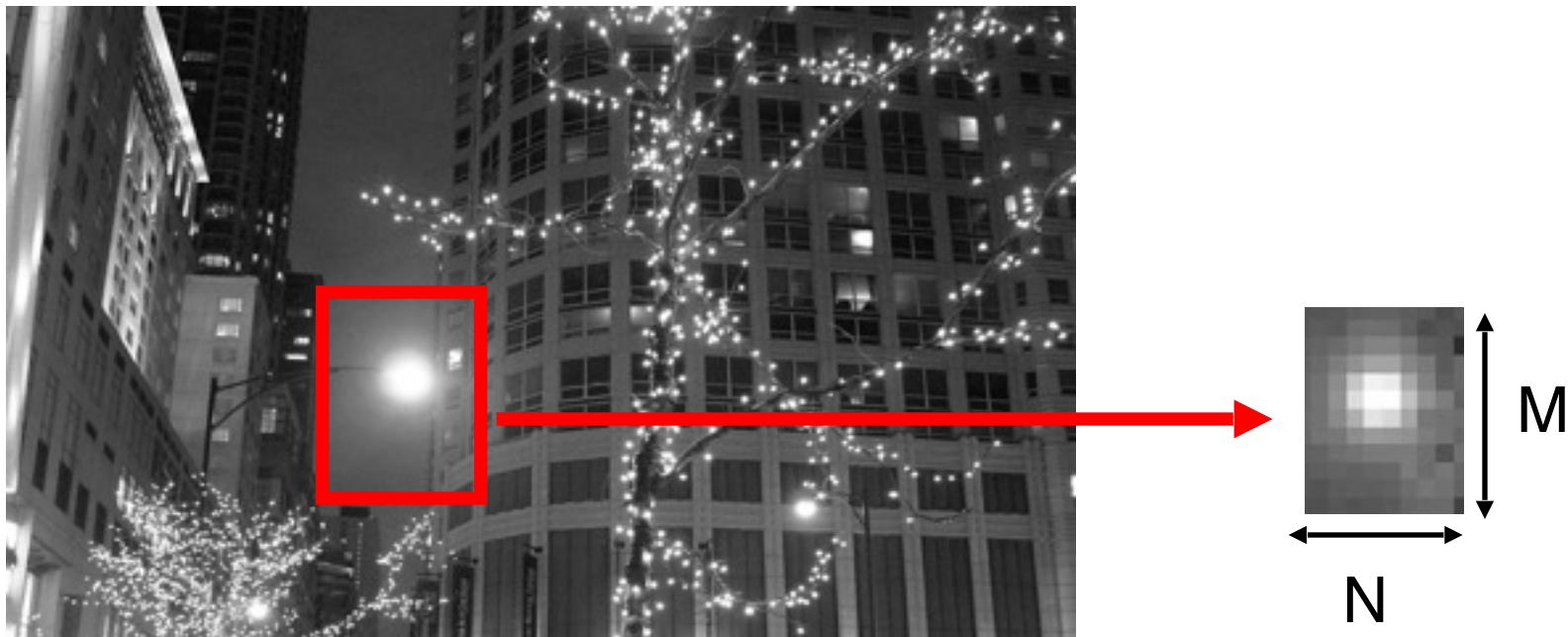
Courtesy of D. Lowe

Change of scale, pose, illumination...

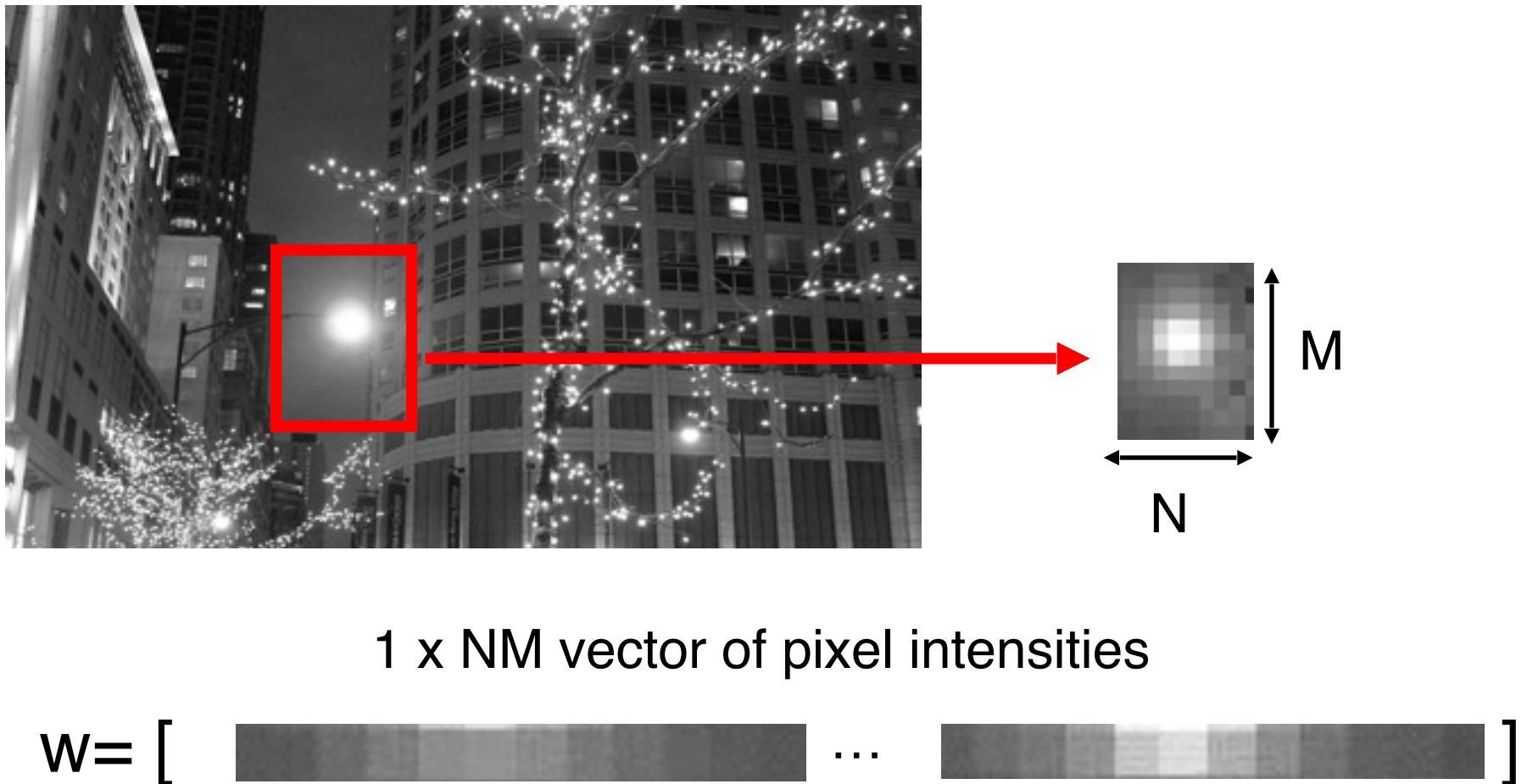
# The simplest descriptor



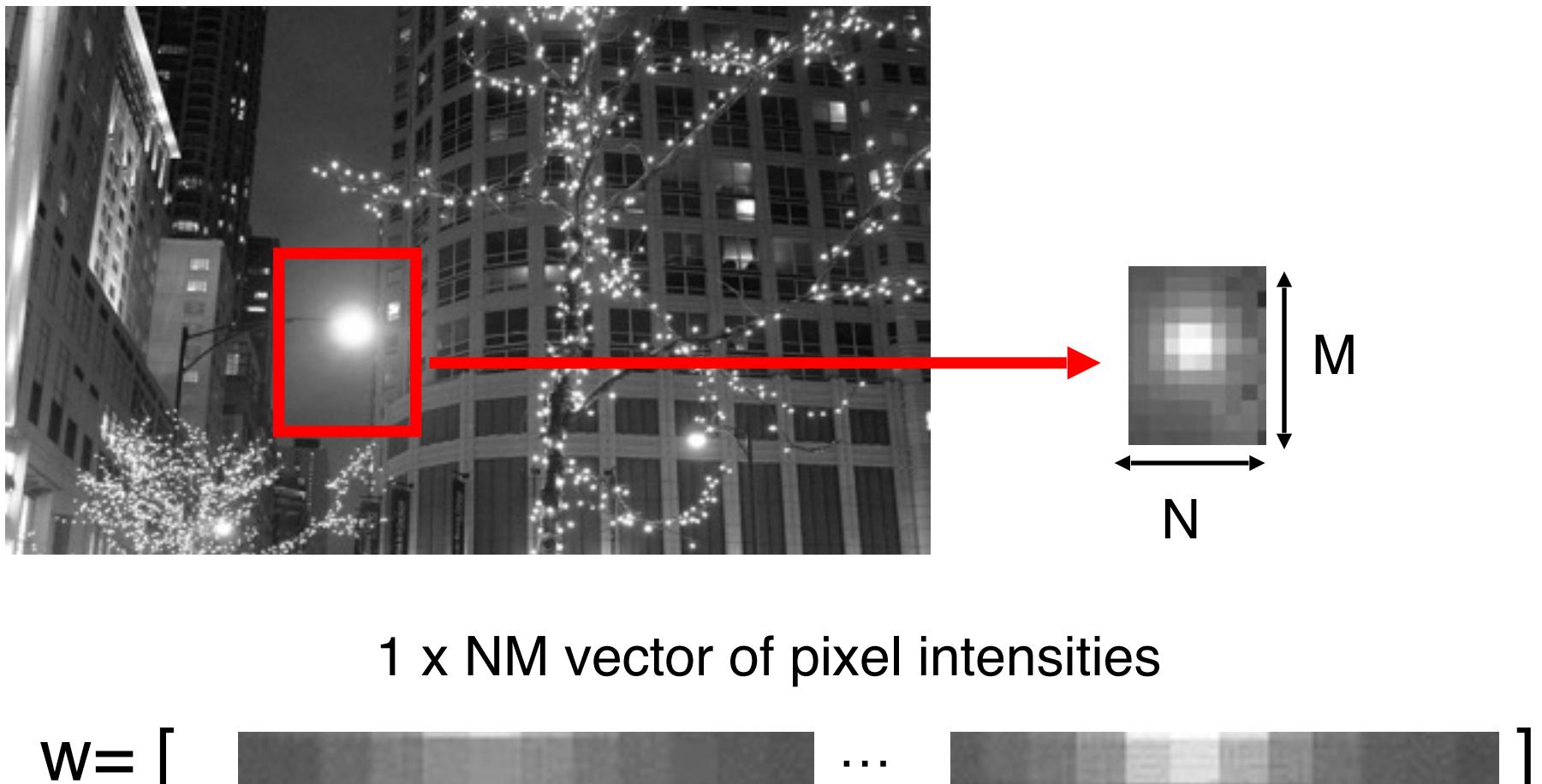
# The simplest descriptor



# The simplest descriptor



# The simplest descriptor



$$w = [ \quad \text{[descriptor image]} \quad \dots \quad \text{[descriptor image]} \quad ]$$

$$w_n = \frac{(w - \bar{w})}{\|(w - \bar{w})\|}$$

Makes the descriptor invariant with respect to affine transformation of the illumination condition

# Why not?

- Sensitive to small variation of:
  - location
  - Pose
  - Scale
  - intra-class variability
- Poorly distinctive

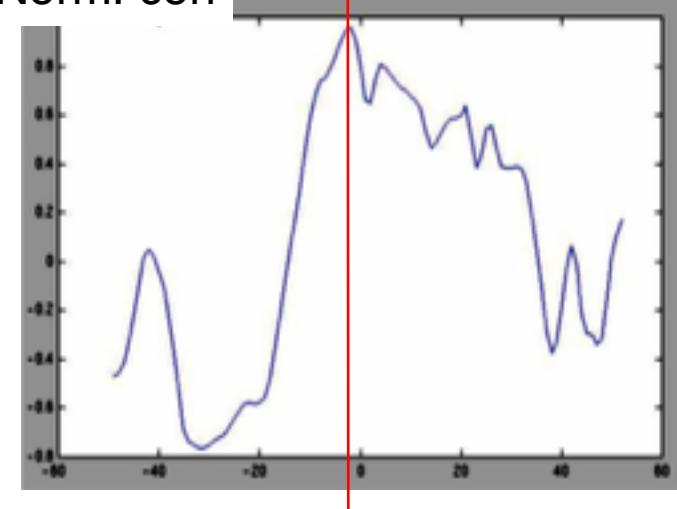
# Sensitive to pose variations



— — — — —



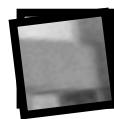
Norm. corr



Normalized Correlation:

$$\mathbf{w}_n \cdot \mathbf{w}'_n = \frac{(\mathbf{w} - \bar{\mathbf{w}})(\mathbf{w}' - \bar{\mathbf{w}}')}{\|(\mathbf{w} - \bar{\mathbf{w}})(\mathbf{w}' - \bar{\mathbf{w}}')\|}$$

# Sensitive to pose variations

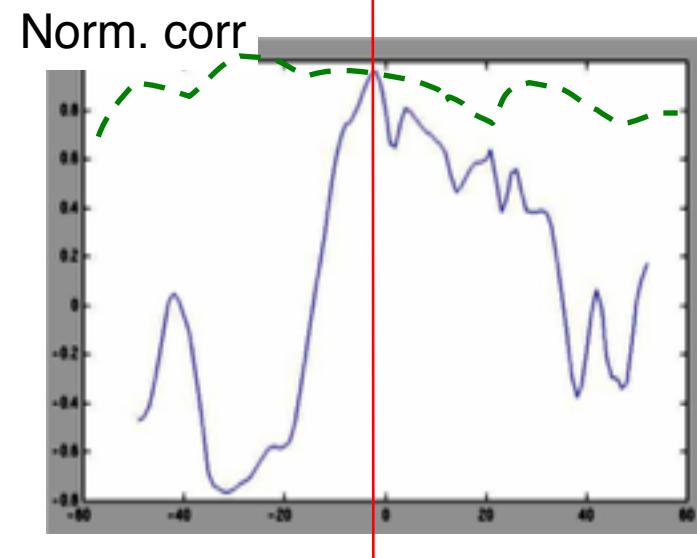


— — — — —



Normalized Correlation:

$$w_n \cdot w'_n = \frac{(w - \bar{w})(w' - \bar{w}')}{\|(w - \bar{w})(w' - \bar{w}')\|}$$

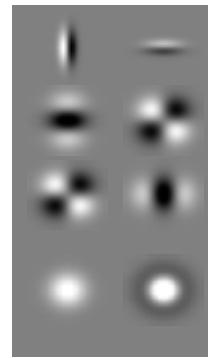


<b>Detector</b>	<b>Illumination</b>	<b>Pose</b>	<b>Intra-class variab.</b>
PATCH	Good	Poor	Poor

# Bank of filters



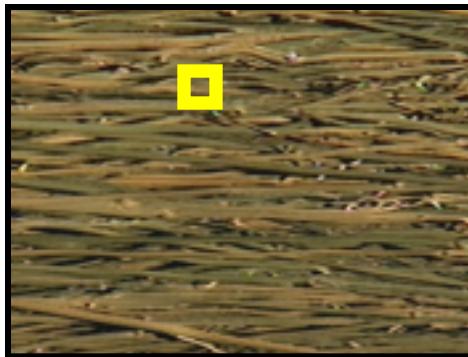
image



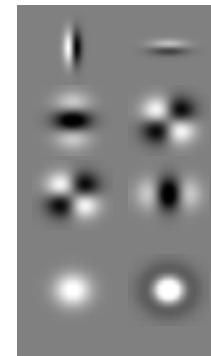
filter bank

More robust but still quite sensitive to pose variations

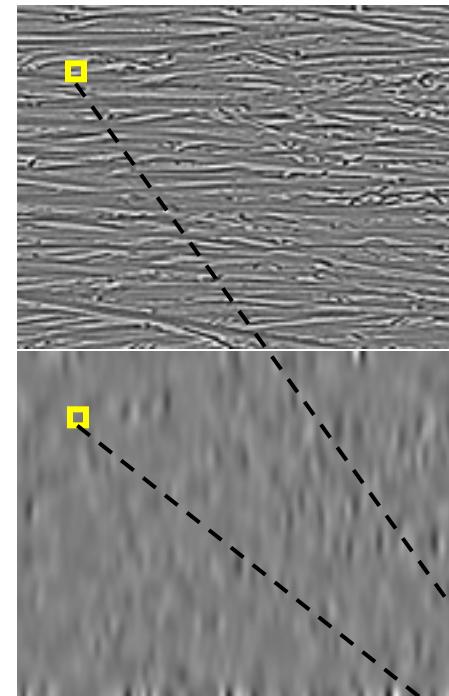
# Bank of filters



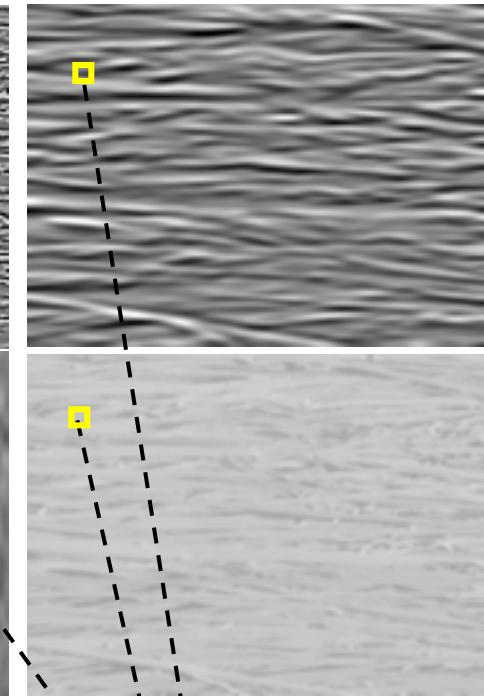
image



filter bank



filter responses



More robust but still quite sensitive to pose variations

<b>Detector</b>	<b>Illumination</b>	<b>Pose</b>	<b>Intra-class variab.</b>
PATCH	Good	Poor	Poor
FILTERS	Good	Medium	Medium

# SIFT descriptor

David G. Lowe. "[Distinctive image features from scale-invariant keypoints.](#)" *IJCV* 60 (2), 04

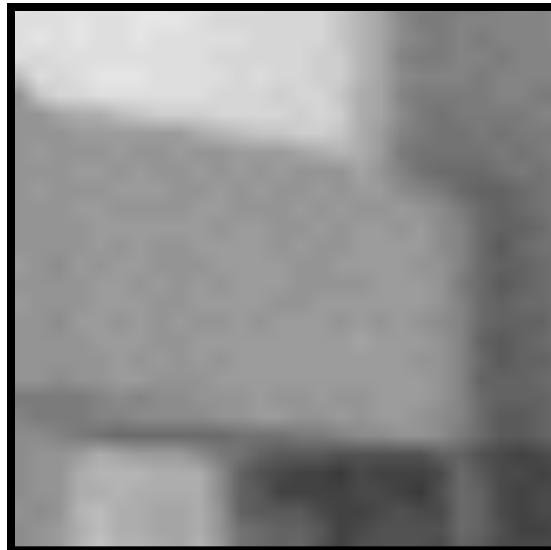
- Alternative representation for image patches
- Location and characteristic scale  $s$  given by DoG detector



# SIFT descriptor

David G. Lowe. "[Distinctive image features from scale-invariant keypoints.](#)" *IJCV* 60 (2), 04

- Alternative representation for image patches
- Location and characteristic scale  $s$  given by DoG detector
  - Compute gradient at each pixel

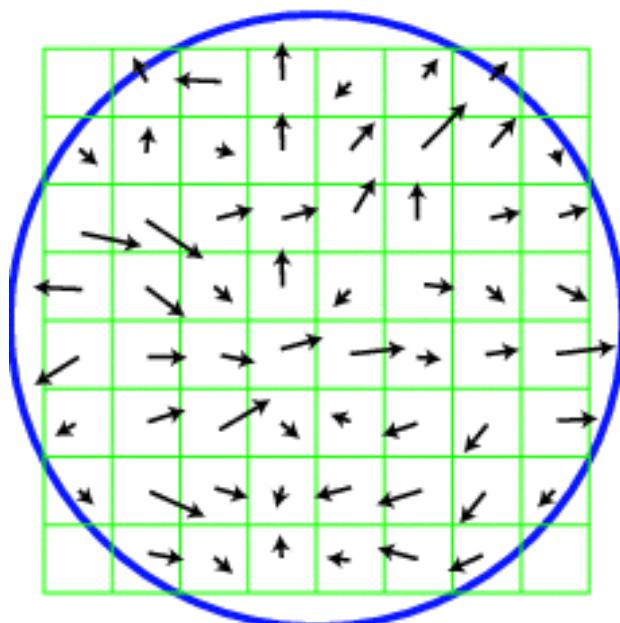


# SIFT descriptor

David G. Lowe. "[Distinctive image features from scale-invariant keypoints.](#)" IJCV 60 (2), 04

- Alternative representation for image patches
- Location and characteristic scale  $s$  given by DoG detector

- Compute gradient at each pixel

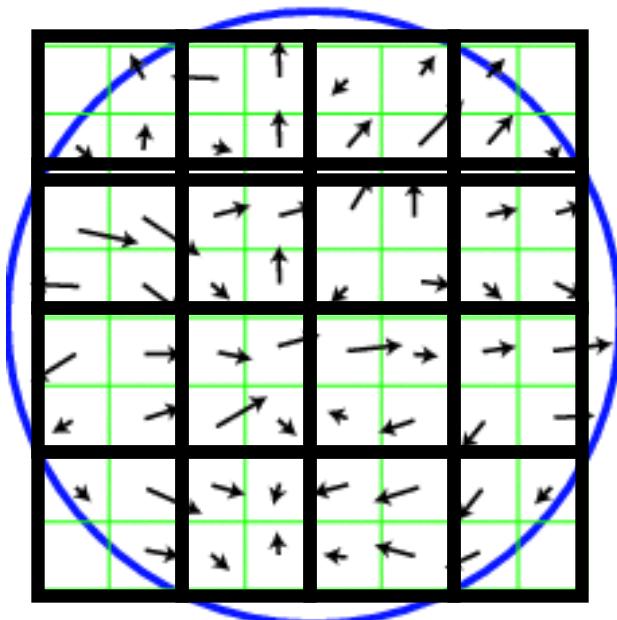


# SIFT descriptor

David G. Lowe. ["Distinctive image features from scale-invariant keypoints."](#) IJCV 60 (2), 04

- Alternative representation for image patches
- Location and characteristic scale  $s$  given by DoG detector

- Compute gradient at each pixel
  - $N \times N$  spatial bins

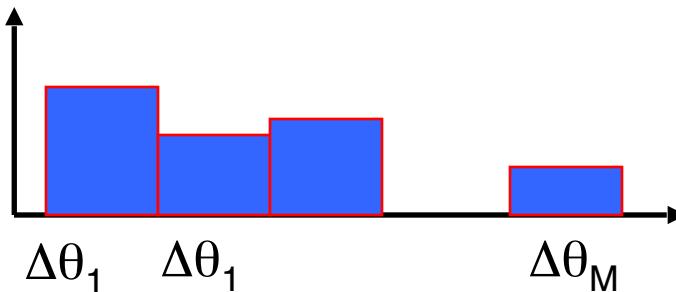
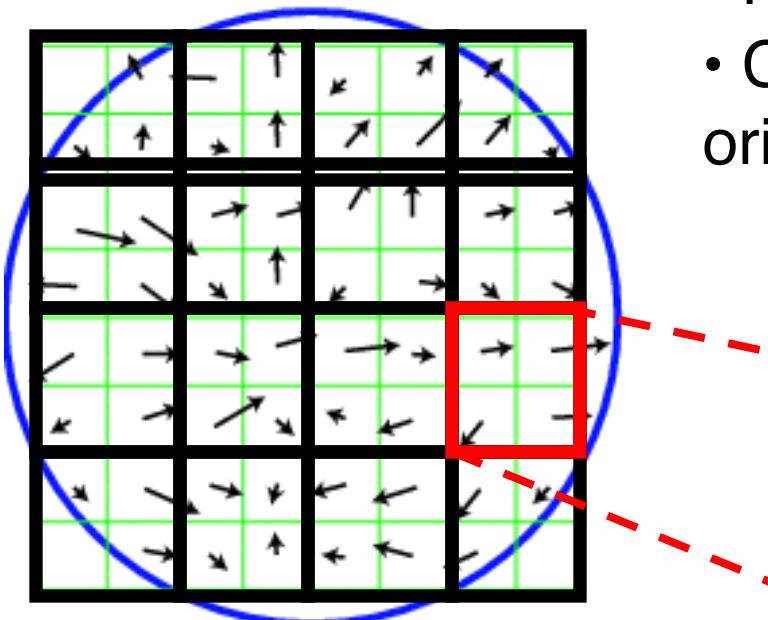


# SIFT descriptor

David G. Lowe. ["Distinctive image features from scale-invariant keypoints."](#) IJCV 60 (2), 04

- Alternative representation for image patches
- Location and characteristic scale  $s$  given by DoG detector

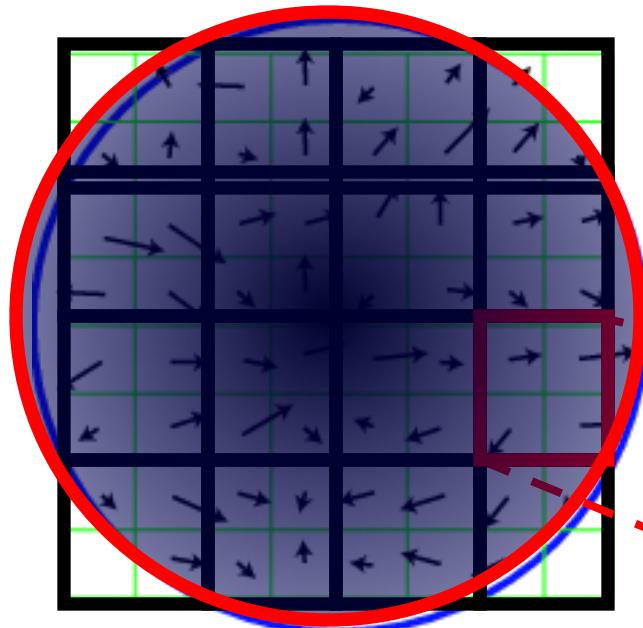
- Compute gradient at each pixel
  - $N \times N$  spatial bins
  - Compute an histogram of  $M$  orientations for each bin



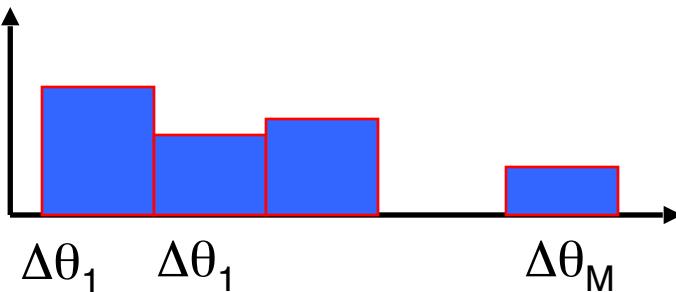
# SIFT descriptor

David G. Lowe. ["Distinctive image features from scale-invariant keypoints."](#) IJCV 60 (2), 04

- Alternative representation for image patches
- Location and characteristic scale  $s$  given by DoG detector



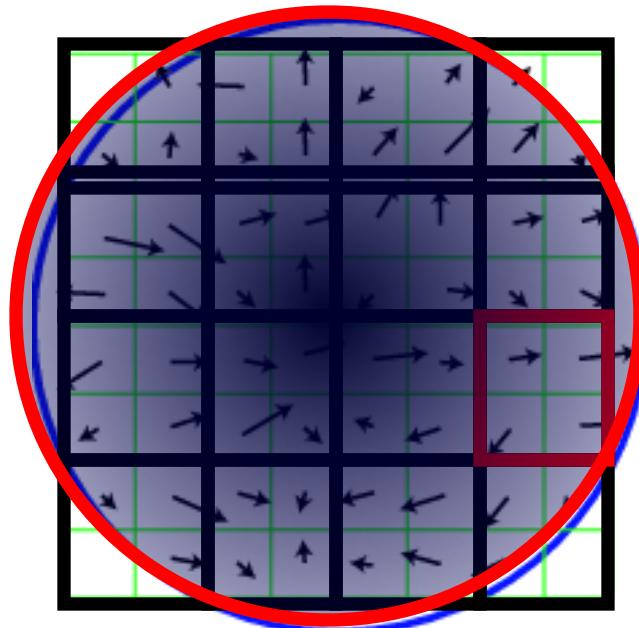
- Compute gradient at each pixel
  - $N \times N$  spatial bins
  - Compute an histogram of  $M$  orientations for each bin
  - Gaussian center-weighting



# SIFT descriptor

David G. Lowe. "[Distinctive image features from scale-invariant keypoints.](#)" IJCV 60 (2), 04

- Alternative representation for image patches
- Location and characteristic scale  $s$  given by DoG detector



- Compute gradient at each pixel
  - $N \times N$  spatial bins
  - Compute an histogram of  $M$  orientations for each bin
  - Gaussian center-weighting
  - Normalized unit norm

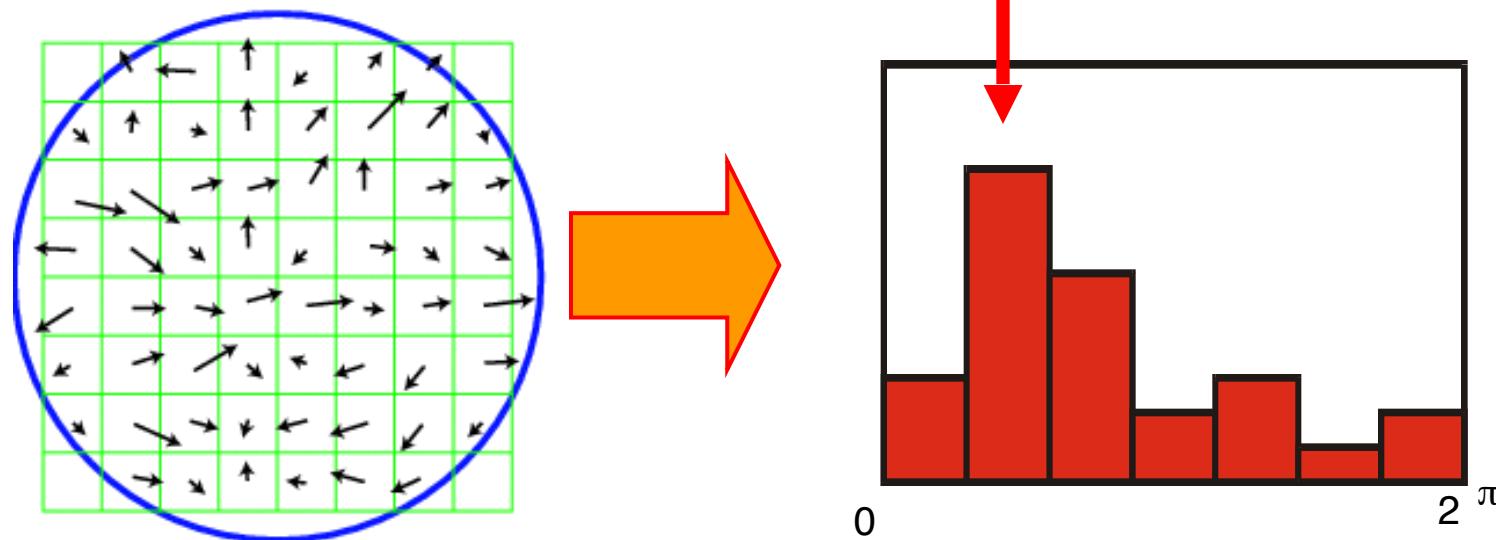
Typically  $M = 8$ ;  $N = 4$   
 $1 \times 128$  descriptor

# SIFT Descriptor

- Robust w.r.t. small variation in:
  - Illumination (thanks to gradient & normalization)
  - Pose (small affine variation thanks to orientation histogram )
  - Scale (scale is fixed by DOG)
  - Intra-class variability (small variations thanks to histograms)

# Rotational Invariance

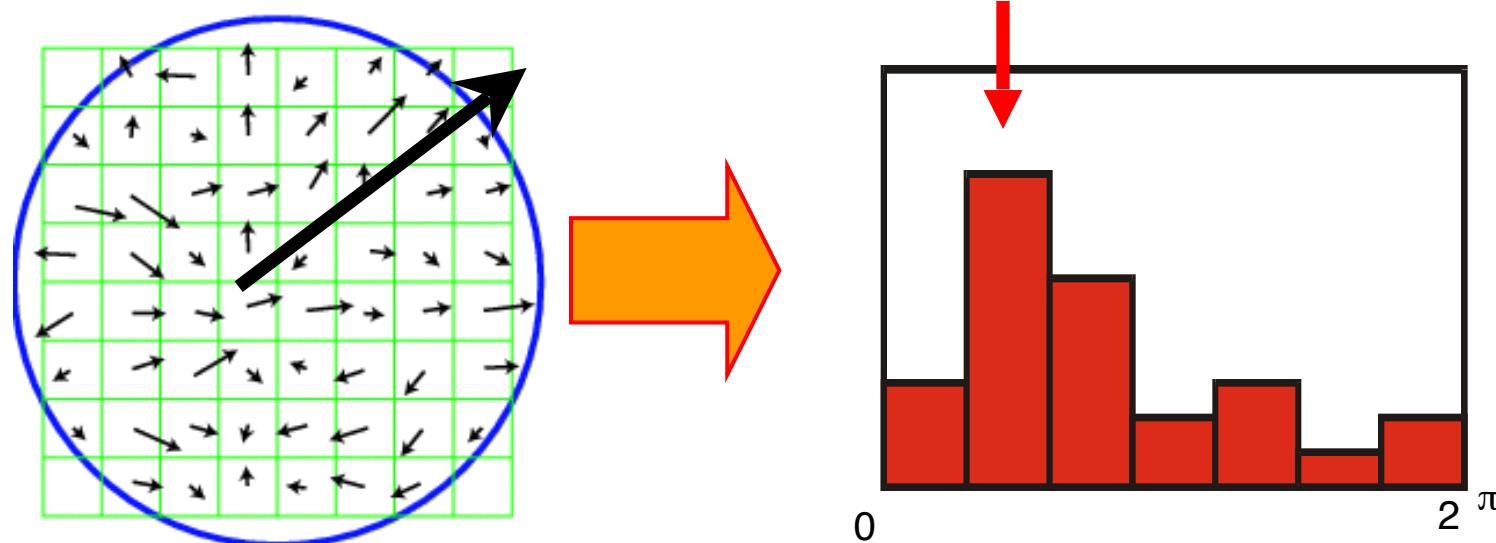
- Find dominant orientation by building smoothed orientation histogram
- Rotate all orientations by the dominant orientation



This makes the SIFT descriptor rotational invariant

# Rotational Invariance

- Find dominant orientation by building smoothed orientation histogram
- Rotate all orientations by the dominant orientation



This makes the SIFT descriptor rotational invariant

# SIFT Rotational Invariance Example



# Rotation invariance (Alternate)

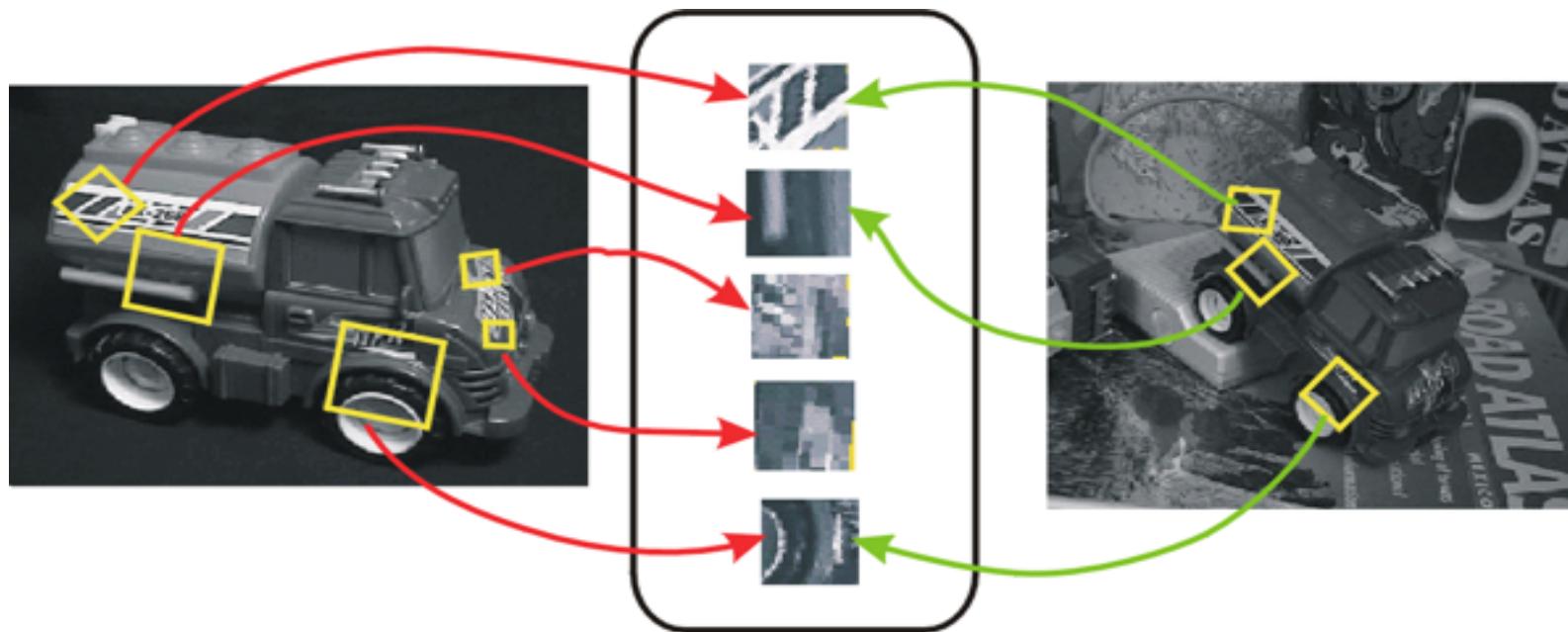
Find dominant orientation of the image patch

- This is given by  $\mathbf{x}_+$ , the eigenvector of  $\mathbf{H}$  corresponding to  $\lambda_+$ 
  - $\lambda_+$  is the *larger* eigenvalue
- Rotate the patch according to this angle



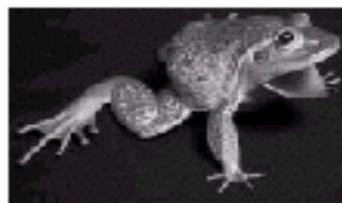
Figure by Matthew Brown

# SIFT Rotational Invariance Example



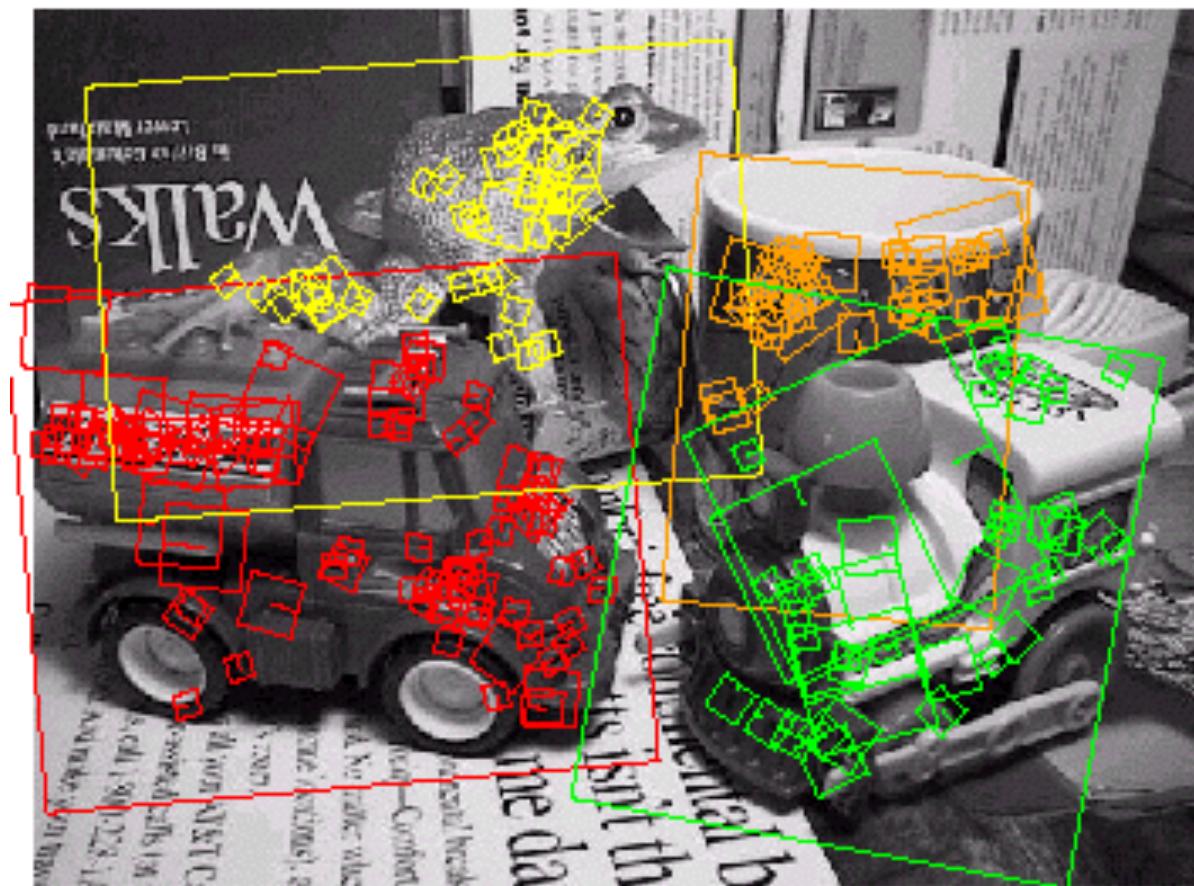
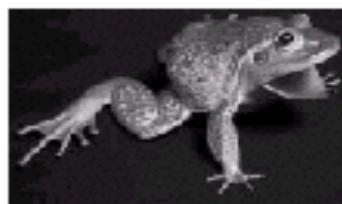
# Matching Using SIFT

David G. Lowe. ["Distinctive image features from scale-invariant keypoints."](#) IJCV 60 (2), 04



# Matching Using SIFT

David G. Lowe. ["Distinctive image features from scale-invariant keypoints."](#) IJCV 60 (2), 04

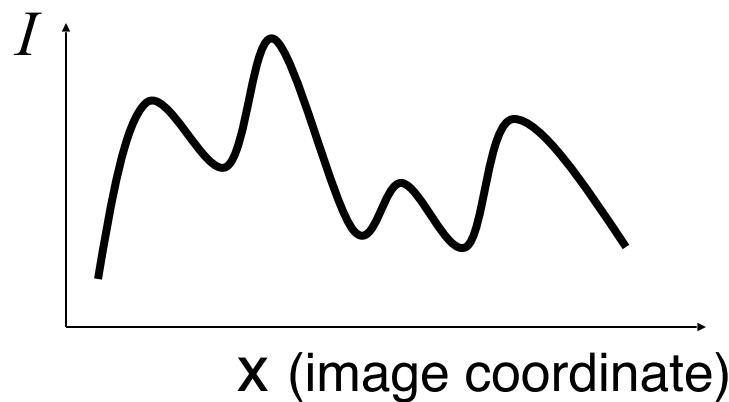


<b>Detector</b>	<b>Illumination</b>	<b>Pose</b>	<b>Intra-class variab.</b>
PATCH	Good	Poor	Poor
FILTERS	Good	Medium	Medium
SIFT	Good	Good	Medium

# Illumination normalization

- *Affine intensity change:*

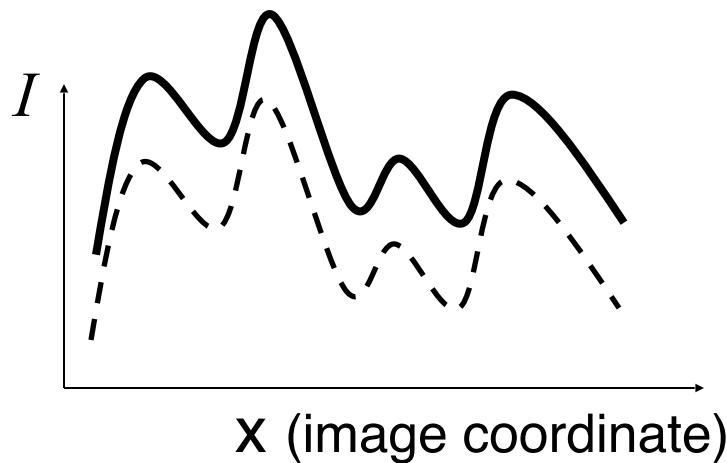
$$I \rightarrow I + b$$



# Illumination normalization

- *Affine intensity change:*

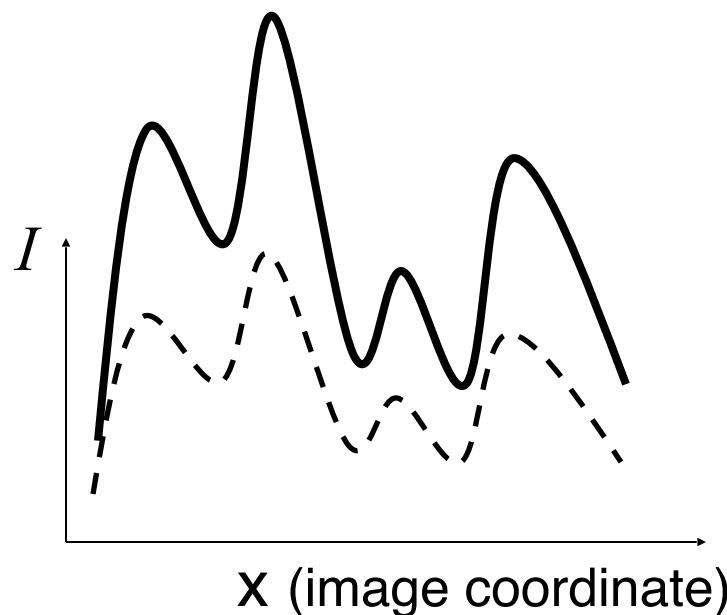
$$I \rightarrow I + b$$



# Illumination normalization

- *Affine intensity change:*

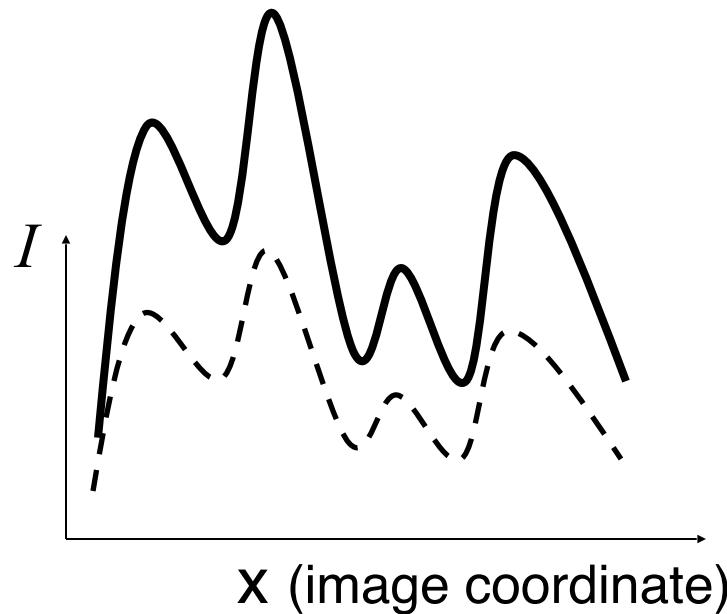
$$\begin{aligned} I &\rightarrow I + b \\ &\rightarrow a I + b \end{aligned}$$



# Illumination normalization

- *Affine intensity change:*

$$\begin{aligned} I &\rightarrow I + b \\ &\rightarrow a I + b \end{aligned}$$



- Make each patch zero mean:

$$\mu = \frac{1}{N} \sum_{x,y} I(x, y)$$

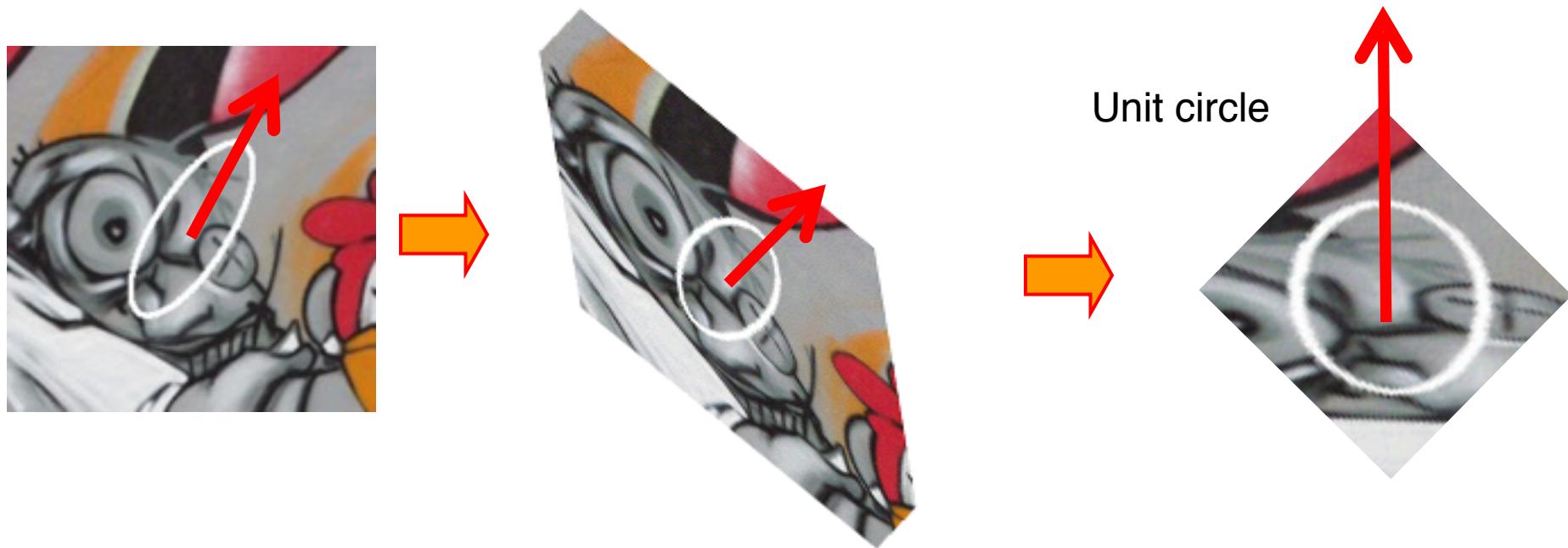
$$Z(x, y) = I(x, y) - \mu$$

- Then make unit variance:

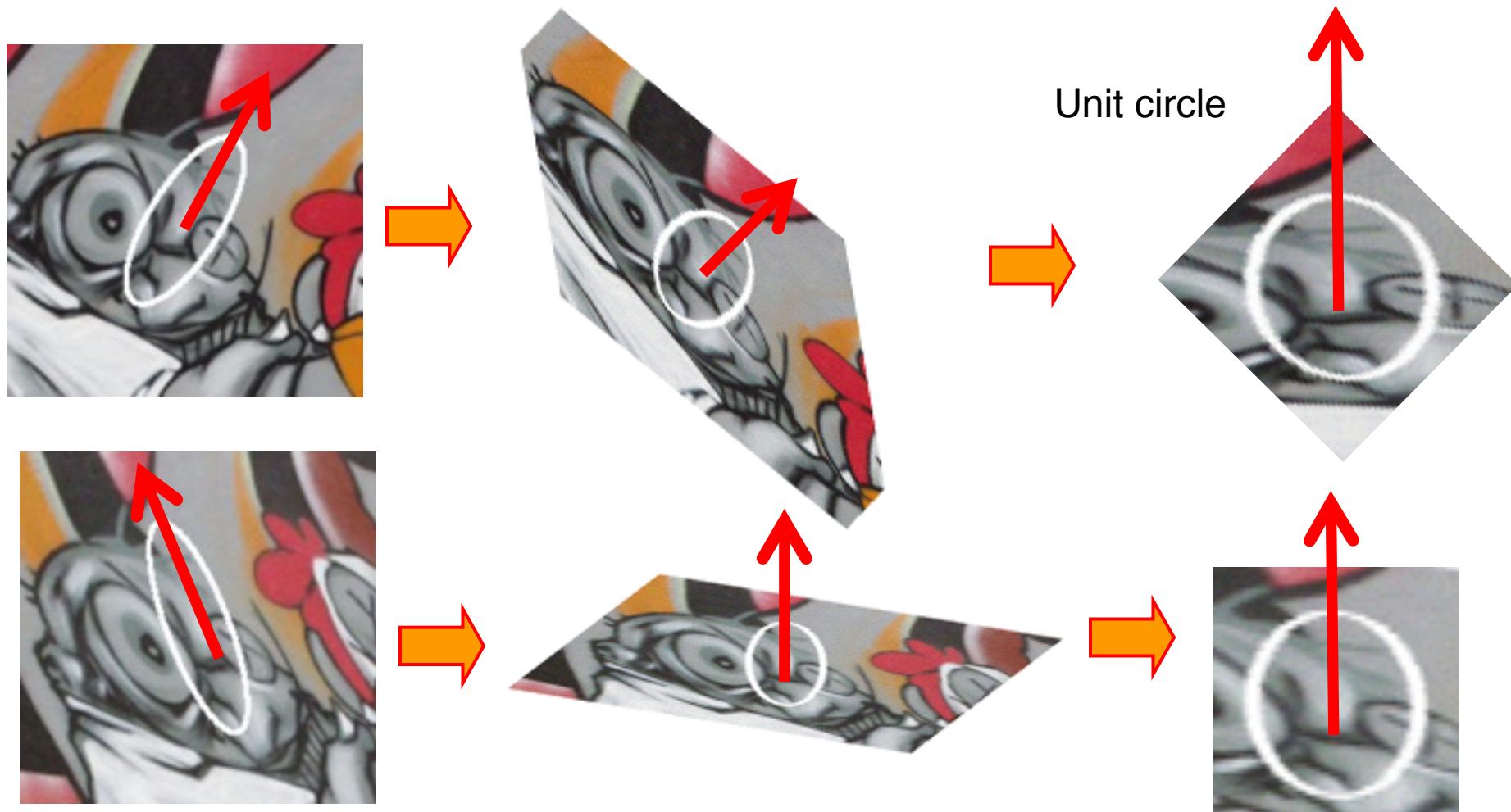
$$\sigma^2 = \frac{1}{N} \sum_{x,y} Z(x, y)^2$$

$$ZN(x, y) = \frac{Z(x, y)}{\sigma}$$

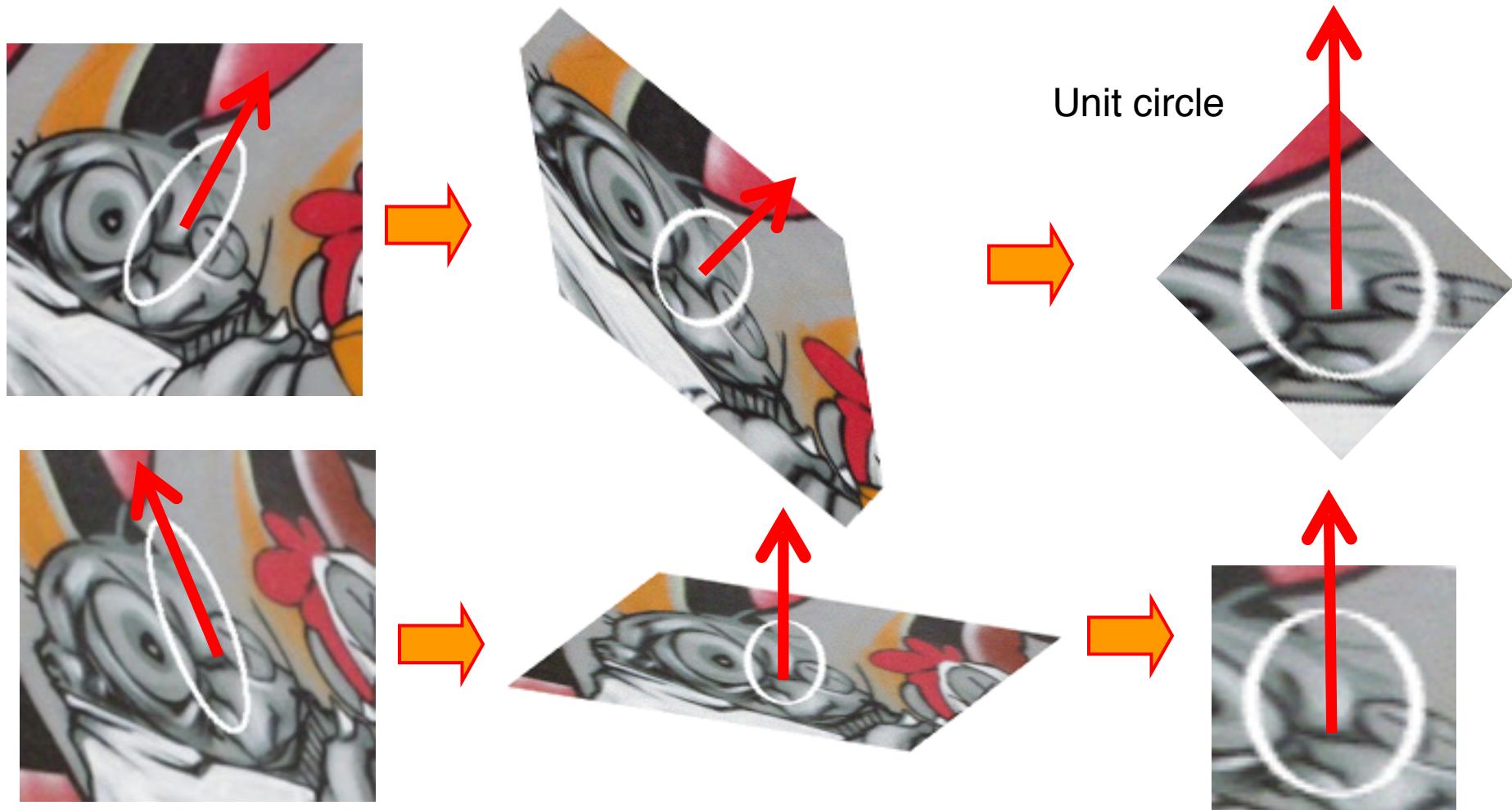
# Pose normalization



# Pose normalization



# Pose normalization



NOTE: location, scale, rotation & affine pose are given by the detector or calculated within the detected regions

# Auxiliary Descriptor Slides

YOU ARE NOT RESPONSIBLE FOR THIS  
MATERIAL AT THIS TIME

If parts of this are reintroduced at later times,  
then you will be responsible for those.

# Open source implementation: [www.vlfeat.org](http://www.vlfeat.org)

The screenshot shows a Google Chrome browser window with the title bar "VLFeat - Tutorials". The address bar contains the URL "www.vlfeat.org/overview/tut.html". The page content is a tutorial overview:

This section features a number of tutorials illustrating some of the main algorithms implemented in VLFeat. The tutorials can be categorized into two classes: the first class of algorithms detect and describe image regions ([features](#)). The second class of algorithms ([cluster](#)) performs clustering on these regions.

**Tutorials**

- Home
- Download
- Documentation
- Tutorials**
  - Covdet
  - HOG
  - SIFT
  - DSIFT/PHOW
  - MSER
  - IKM
  - HIKM
  - AIB
  - Quick shift
  - SLIC
  - kd-tree
  - Distance transf.
  - Utils
  - Pegasos
  - Plots: rank
- Applications

**Features**

- [Covariant detectors](#). An introduction to computing co-variant features like Harris-Affine.
- [Histogram of Oriented Gradients \(HOG\)](#). Getting started with this ubiquitous representation for object recognition.
- [Scale Invariant Feature Transform \(SIFT\)](#). Getting started with this popular feature detector / descriptor.
- [Dense SIFT \(DSIFT\) and PHOW](#). A state-of-the-art descriptor for image categorization.
- [Maximally Stable Extremal Regions \(MSER\)](#). Extracting MSERs from an image.
- [Image distance transform](#). Compute the image distance transform for fast part models and edge matching.

**Clustering**

- [Integer optimized k-means \(IKM\)](#). A quick overview of VLFeat fast  $k$ -means implementation.
- [Hierarchical k-means \(HIKM\)](#). Create a fast  $k$ -means tree for integer data.
- [Agglomerative Information Bottleneck \(AIB\)](#). Cluster discrete data based on the mutual information between the data and cluster labels.
- [Quick shift](#). An introduction which shows how to create superpixels using this quick mode seeking method.
- [SLIC](#). An introduction to SLIC superpixels.

**Other**

- [Pegasos SVM](#). Learn a binary classifier and check its convergence plotting the energy value.
- [Forests of kd-trees](#). Approximate nearest neighbor queries in high dimensions using an optimized forest of kd-trees.
- [Plotting functions for rank evaluation](#). Learn how to plot ROC, DET, and precision-recall curves.
- [MATLAB Utilities](#). A list of useful MATLAB functions bundled with VLFeat.

# **Video Detectors / Features**

# STIP: Space-Time Interest Points

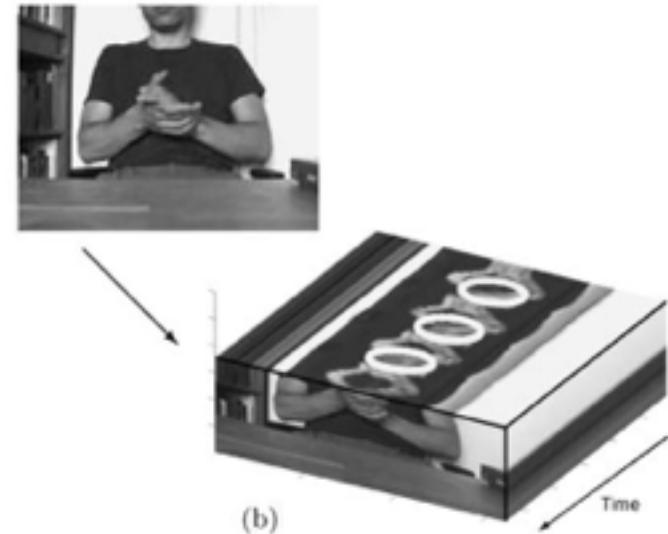
Source: Laptev. "On Space-Time Interest Points." Intl Journal of Computer Vision. 64(2/3):107-123. 2005.

- Basic idea is to detect points in the video that have significant local variations in both space and time.
- Builds on the existing work of Harris corner detector and incorporates a scale parameter.

$$\mu = g(\cdot; \sigma_i^2, \tau_i^2) * \begin{pmatrix} L_x^2 & L_x L_y & L_x L_t \\ L_x L_y & L_y^2 & L_y L_t \\ L_x L_t & L_y L_t & L_t^2 \end{pmatrix}$$

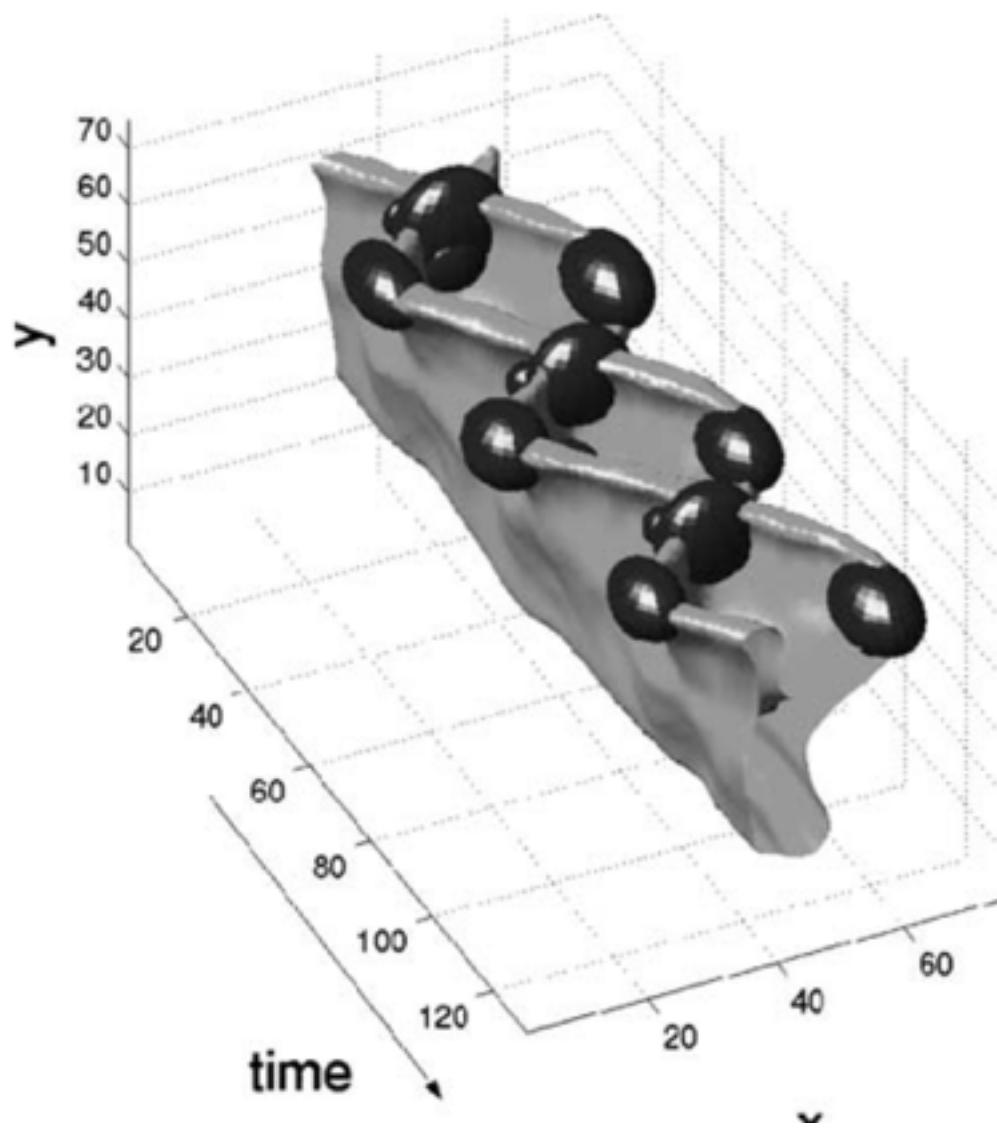
$$H = \det(\mu) - k \operatorname{trace}^2(\mu)$$

- The original work incorporates a scale-selection term; most subsequent works densely sample scale.



# STIP: Space-Time Interest Points

Source: Laptev. "On Space-Time Interest Points." Intl Journal of Computer Vision. 64(2/3):107-123. 2005.



# STIP: Space-Time Interest Points

Source: Laptev. "On Space-Time Interest Points." Intl Journal of Computer Vision. 64(2/3):107-123. 2005.



Video from Laptev's CVPR 2008 slides.

# STIP: Space-Time Interest Points

Source: Laptev. "On Space-Time Interest Points." Intl Journal of Computer Vision. 64(2/3):107-123. 2005.



Video from Laptev's CVPR 2008 slides.

# STIP: Space-Time Interest Points

Source: Laptev. "On Space-Time Interest Points." Intl Journal of Computer Vision. 64(2/3):107-123. 2005.



Video from Laptev's CVPR 2008 slides.

# Dollár's Cuboids

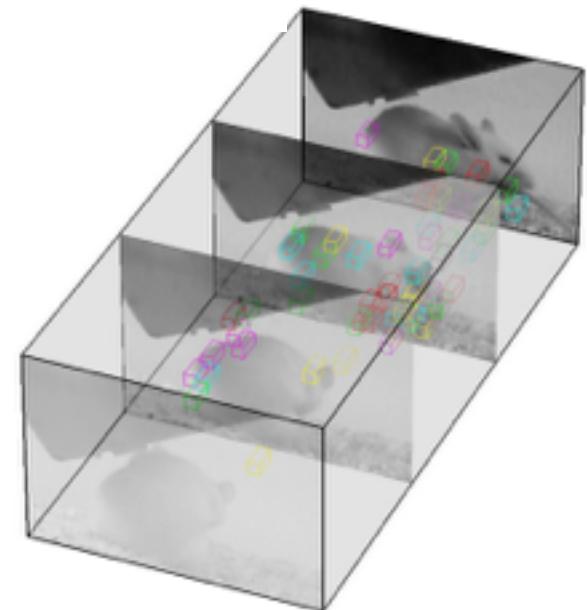
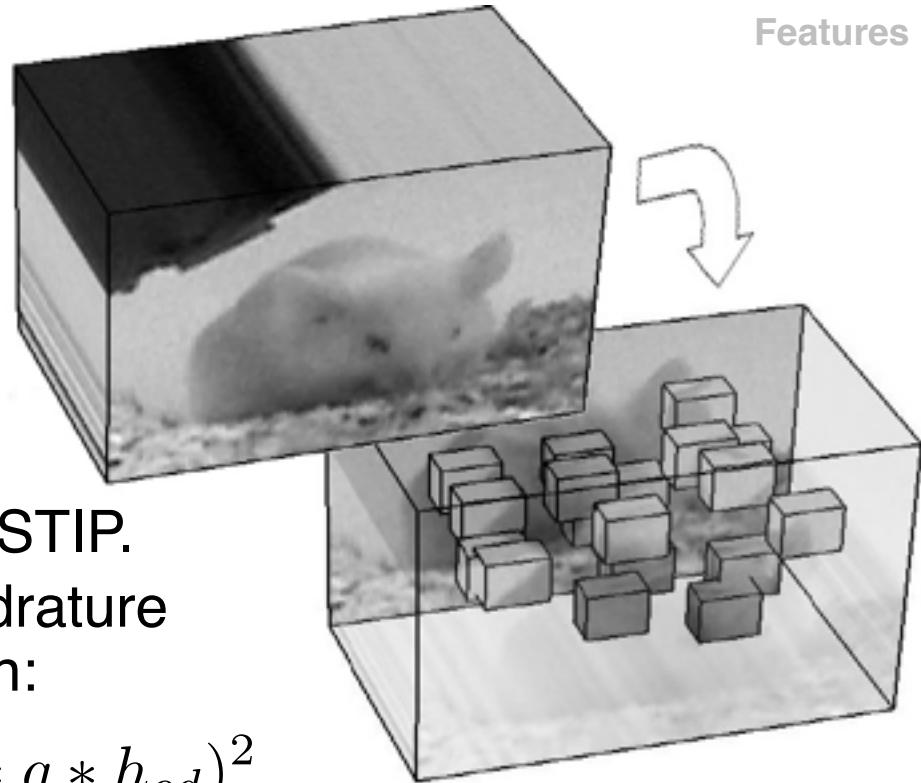
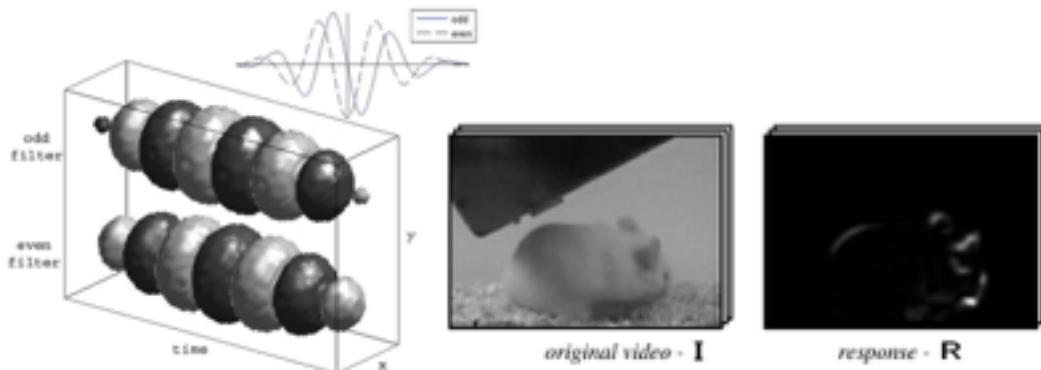
Source: Dollar et al. "Behavior Recognition" ICCV PETS Workshop 2005.

- Detector fires when local image intensities contain periodic frequency components.
- It will fire more frequently than STIP.
- Based on temporal Gabor quadrature pair filter with response function:

$$R = (I * g * h_{ev})^2 + (I * g * h_{od})^2$$

$$h_{ev}(t; \tau) = -\cos(8\pi t/\tau) e^{-t^2/\tau^2}$$

$$h_{od}(t; \tau) = -\sin(8\pi t/\tau) e^{-t^2/\tau^2}$$



# Dense Sampling of Locations

- Motivated by successes in object recognition where densely sampled features outperformed sparse ones, it has become common to sample densely for activity recognition too.
- Example videos below
  - 7x7x7 non-overlapping samples,
  - Simple temporal derivative (much simpler than HOF and HOG3D).
  - k-Means in 128 *visual words*.



# Dense Sampling of Locations

- Motivated by successes in object recognition where densely sampled features outperformed sparse ones, it has become common to sample densely for activity recognition too.
- Example videos below
  - 7x7x7 non-overlapping samples,
  - Simple temporal derivative (much simpler than HOF and HOG3D).
  - k-Means in 128 *visual words*.



# Dense Sampling of Locations

- Motivated by successes in object recognition where densely sampled features outperformed sparse ones, it has become common to sample densely for activity recognition too.
- Example videos below
  - 7x7x7 non-overlapping samples,
  - Simple temporal derivative (much simpler than HOF and HOG3D).
  - k-Means in 128 *visual words*.



# Dense Sampling of Locations

- Motivated by successes in object recognition where densely sampled features outperformed sparse ones, it has become common to sample densely for activity recognition too.
- Example videos below
  - 7x7x7 non-overlapping samples,
  - Simple temporal derivative (much simpler than HOF and HOG3D).
  - k-Means in 128 *visual words*.



# Discussion: Local Spatiotemporal Features

- Benefits of local feature methods:
  - Robustness to viewpoint changes and occlusion.
  - Relatively computationally inexpensive.
  - Do not need to detect and track the agent.
  - Implicitly incorporate motion, form, and context.

# Discussion: Local Spatiotemporal Features

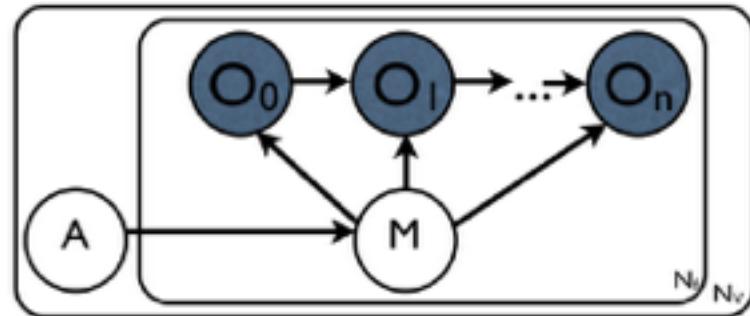
- Benefits of local feature methods:
  - Robustness to viewpoint changes and occlusion.
  - Relatively computationally inexpensive.
  - Do not need to detect and track the agent.
  - Implicitly incorporate motion, form, and context.
- But, they may be too limited for comprehensive activity recognition.
  - Temporal structure is diminished or lost.
  - Human performance suggests a broader spatial and temporal range may be needed for good activity recognition.
  - Typically do not incorporate any inter-relationships among the extracted features or points.

# Trajectories by Local Keypoint Tracking

Source: Messing et al. "Activity Recognition using velocity histories of tracked keypoints." ICCV 2009.

- Detects corners in the image and tracks them using a KLT tracker.
  - 500 points at a time w/ replacement.
  - Mean duration is 150 frames.
- Represent trajectories by quantized trajectory velocity.
- Learn a mixture model over velocity Markov chains.
- Each action has a distribution over the mixture components.
- Joint model over action and observations:
- Learn via EM.

Method	Percent Correct
Temporal Templates [6]	33
Spatio-Temporal Cuboids [7]	36
Space-Time Interest Points [12]	59
Velocity Histories (Sec. 3)	63
Latent Velocity Histories (Sec. 7)	67
Augmented Velocity Histories (Sec. 6)	89



$$\begin{aligned}
 P(A, O) &= \sum_M P(A, M, O) = \\
 P(A) \prod_f^{N_f} \sum_i^{N_m} P(M_f^i | A) P(O_{0,f} | M_f^i) \\
 &\prod_{t=1}^{T_f} P(O_{t,f} | O_{t-1,f}, M_f^i)
 \end{aligned}$$

# Trajectories by Local Keypoint Tracking

Source: Messing et al. "Activity Recognition using velocity histories of tracked keypoints." ICCV 2009.



# Trajectories by Local Keypoint Tracking

Source: Messing et al. "Activity Recognition using velocity histories of tracked keypoints." ICCV 2009.



# Trajectories by Local Keypoint Tracking

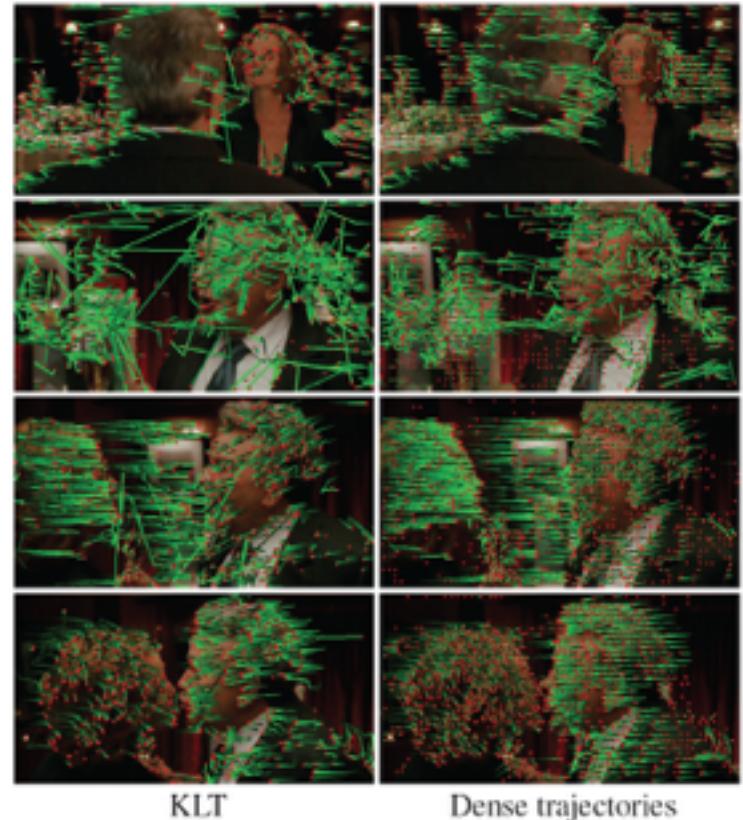
Source: Messing et al. "Activity Recognition using velocity histories of tracked keypoints." ICCV 2009.



# Dense Trajectories

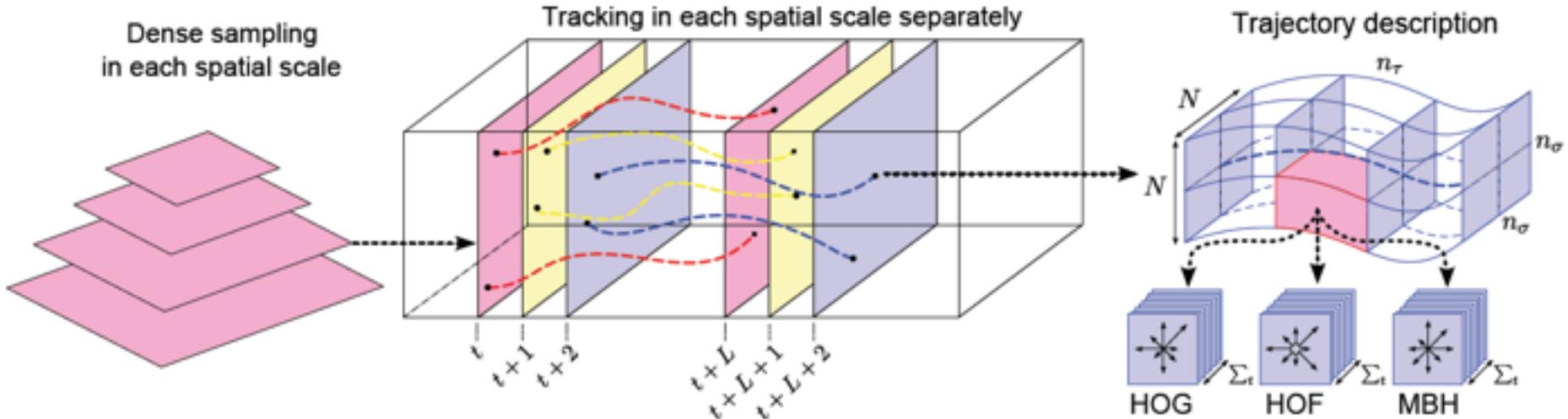
Source: Wang et al. "Action Recognition by Dense Trajectories." CVPR 2011.

- Dense sampling improves object recognition and action recognition; why not use it for trajectories?
- Matching features across frames is very expensive.
- Proposes a method to track the trajectories densely using a single dense optical flow field calculation.
  - Global smoothness enforced.
- Compute the descriptors aligned with the trajectories using HOG/HOF/MBH.



KLT

Dense trajectories



# Dense Trajectories: Convincing Improvements

Source: Wang et al. "Action Recognition by Dense Trajectories." CVPR 2011.

	KTH		YouTube		Hollywood2		UCF sports	
	KLT	Dense trajectories						
Trajectory	88.4%	90.2%	58.2%	67.2%	46.2%	47.7%	72.8%	75.2%
HOG	84.0%	86.5%	71.0%	74.5%	41.0%	41.5%	80.2%	83.8%
HOF	92.4%	93.2%	64.1%	72.8%	48.4%	50.8%	72.7%	77.6%
MBH	93.4%	95.0%	72.9%	83.9%	48.6%	54.2%	78.4%	84.8%
Combined	<b>93.4%</b>	<b>94.2%</b>	<b>79.9%</b>	<b>84.2%</b>	<b>54.6%</b>	<b>58.3%</b>	<b>82.1%</b>	<b>88.2%</b>

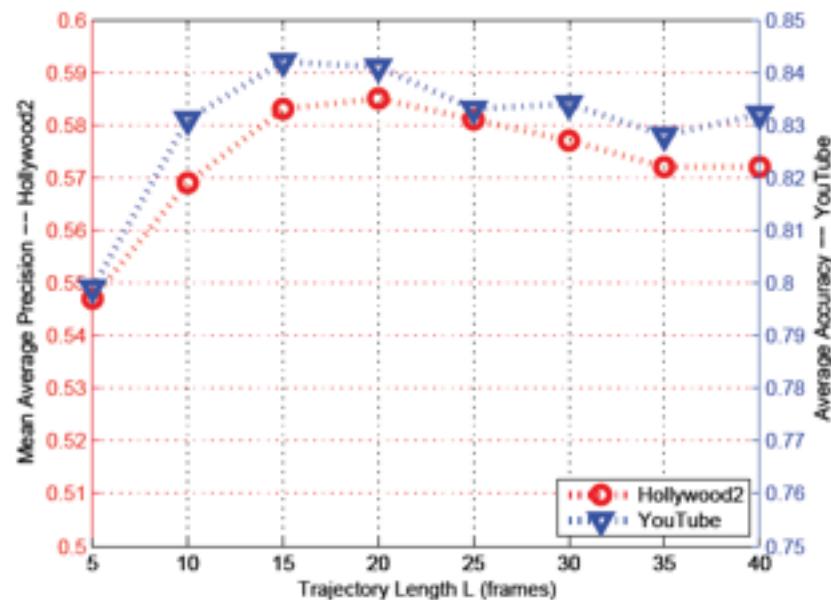
Table 1. Comparison of KLT and dense trajectories as well as different descriptors on KTH, YouTube, Hollywood2 and UCF sports. We report average accuracy over all classes for KTH, YouTube and UCF sports and mean AP over all classes for Hollywood2.

	KTH	YouTube	Hollywood2	UCF sports
Laptev <i>et al.</i> [14]	91.8%	Liu <i>et al.</i> [16]	71.2%	Wang <i>et al.</i> [32]
Yuan <i>et al.</i> [35]	93.3%	Ikizler-Cinbis <i>et al.</i> [9]	75.21%	Gilbert <i>et al.</i> [8]
Gilbert <i>et al.</i> [8]	94.5%			Ullah <i>et al.</i> [31]
Kovashka <i>et al.</i> [12]	<b>94.53%</b>			Taylor <i>et al.</i> [29]
Our method	94.2%	Our method	<b>84.2%</b>	Our method
			<b>58.3%</b>	Our method
				<b>88.2%</b>

Table 2. Comparison of our dense trajectories characterized by our combined descriptor (Trajectory+HOG+HOF+MBH) with state-of-the-art methods in the literature.

	KLT	Dense trajectories	Ikizler-Cinbis [9]
b_shoot	34.0%	43.0%	<b>48.48%</b>
bike	87.6%	<b>91.7%</b>	75.17%
dive	<b>99.0%</b>	<b>99.0%</b>	95.0%
golf	95.0%	<b>97.0%</b>	95.0%
h_ride	76.0%	<b>85.0%</b>	73.0%
s_juggle	65.0%	<b>76.0%</b>	53.0%
swing	86.0%	<b>88.0%</b>	66.0%
t.swing	71.0%	71.0%	<b>77.0%</b>
t.jump	93.0%	<b>94.0%</b>	93.0%
v.spike	<b>96.0%</b>	95.0%	85.0%
walk	76.4%	<b>87.0%</b>	66.67%
Accuracy	79.9%	<b>84.2%</b>	75.21%

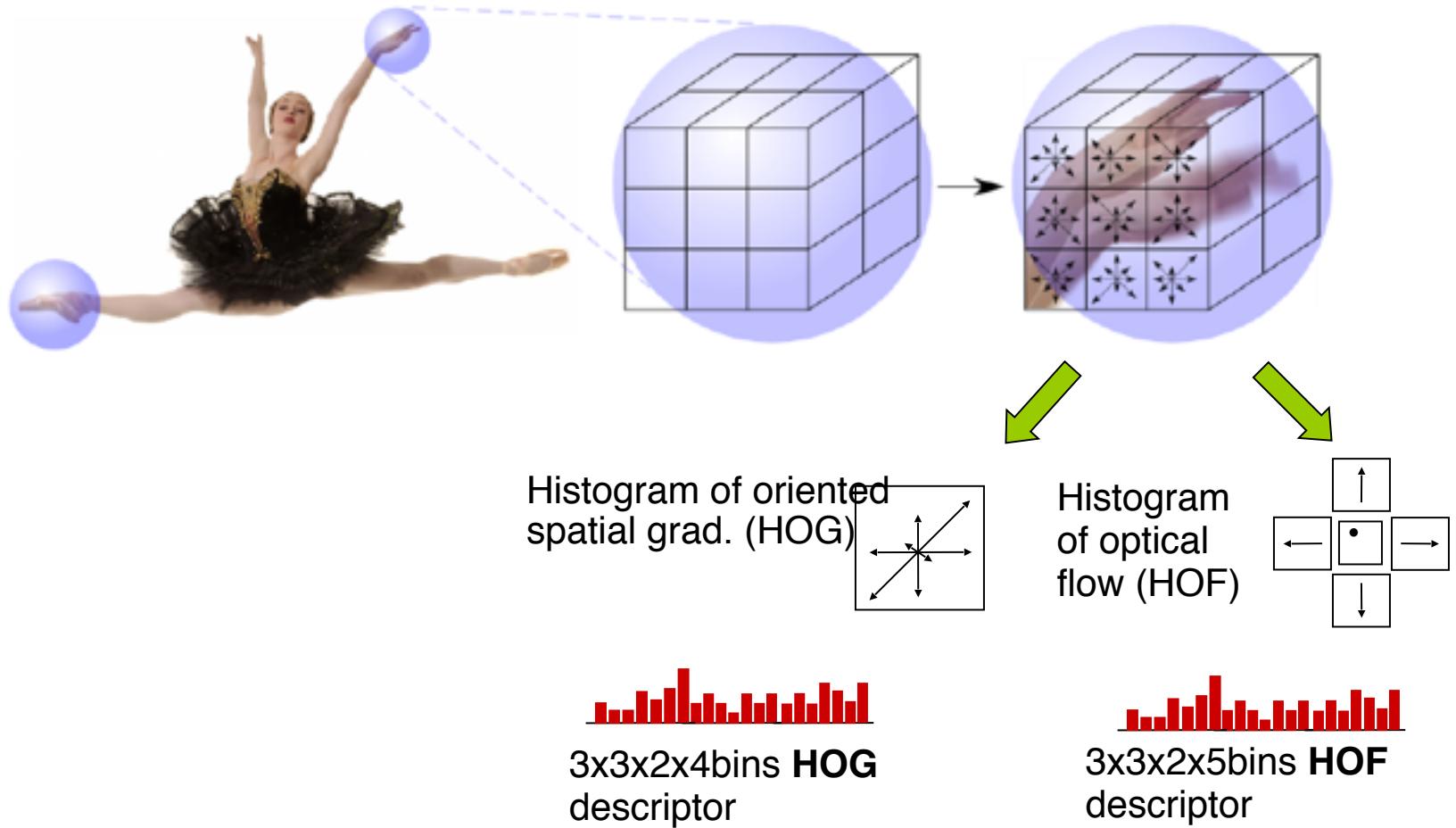
Table 3. Accuracy per action class for the YouTube dataset. We compare with the results reported in [9].



# Local Descriptors: HOG/HOF

Source: materials adapted from Laptev's CVPR 2008 slides.

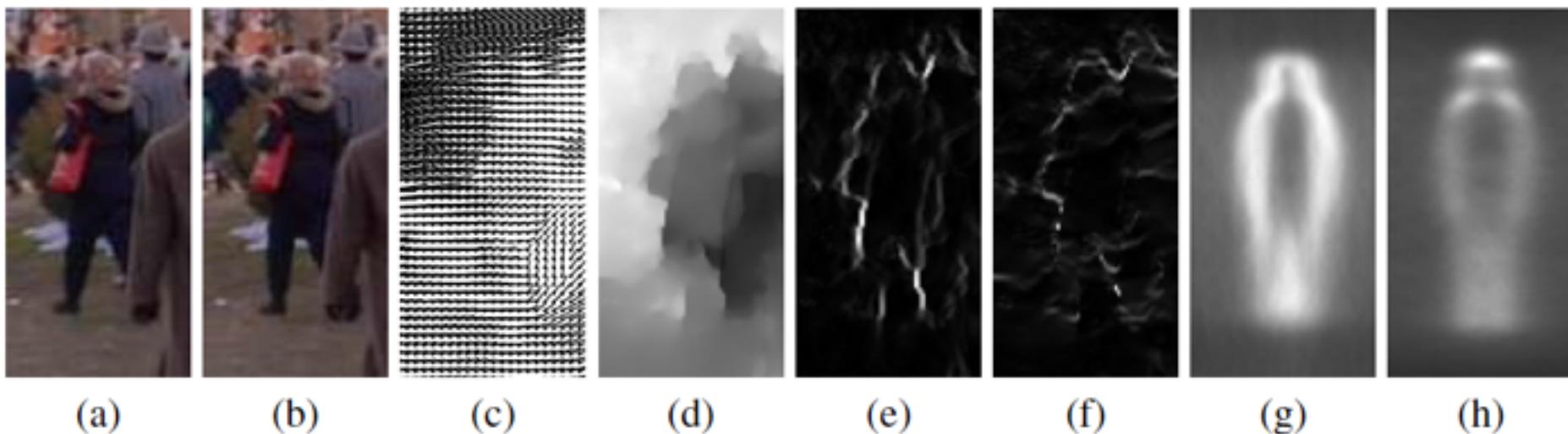
Description (sparse/dense) in space-time patches.



# Motion Boundary Histograms

Source: Dalal et al. "Human Detection Using Oriented Histograms of Flow and Appearance." ECCV 2006.

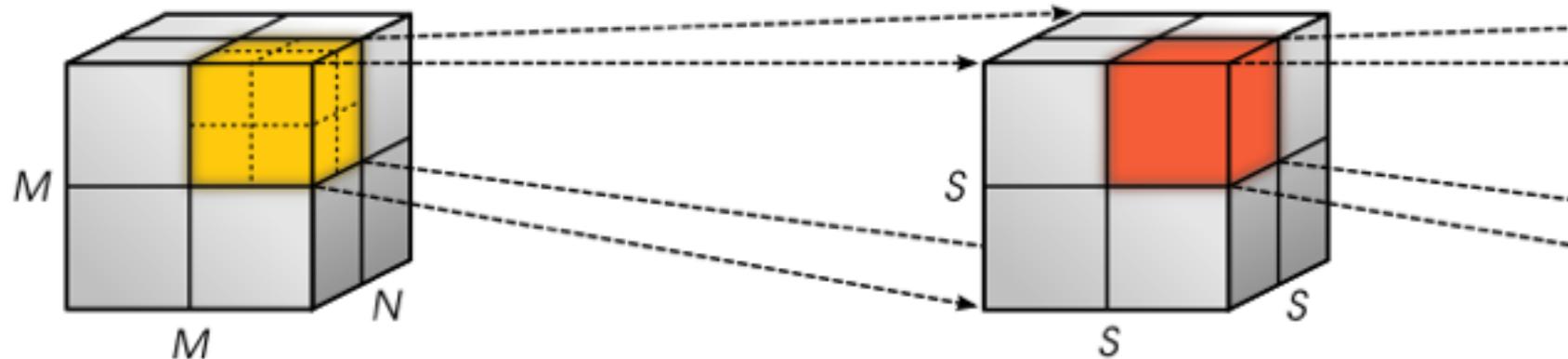
- Rather than HOF directly, MBH focuses on histograms of differential optical flow.
  - Descriptive of motion articulation but resistant to background and camera motion.
- Compute optical flow and take differentials separately over dx and dy. Use separate histograms over resulting dx and dy images.



**Fig. 3.** Illustration of the MBH descriptor. (a,b) Reference images at time  $t$  and  $t+1$ . (c,d) Computed optical flow, and flow magnitude showing motion boundaries. (e,f) Gradient magnitude of flow field  $\mathcal{I}^x$ ,  $\mathcal{I}^y$  for image pair (a,b). (g,h) Average MBH descriptor over all training images for flow field  $\mathcal{I}^x$ ,  $\mathcal{I}^y$ .

# Local Descriptors: HOG3D

Source: Kläser et al. "A Spatio-Temporal Descriptor Based on 3-D Gradients." BMVC 2008. And the provided poster.



## (i) Full descriptor

- ✓ Descriptor for a local *support region* around 3D position in the video
- ✓ The support region is divided into a set of  $M \times M \times N$  cells
- ✓ For each cell, an orientation histogram is computed
- ✓ All cell histograms are concatenated
- ✓ Final vector is normalized and values are limited to a given *cut-off value*

## (ii) Histogram of gradient orientations

- ✓ A histogram of gradient orientations is computed over a set of gradients
- ✓ Therefore, a given cell is divided into  $S \times S \times S$  subblocks
- ✓ For each subblock, its mean gradient is computed and quantized
- ✓ Sum over all quantized gradients in subblocks give the histogram

# Local Descriptors: HOG3D

Source: Kläser et al. "A Spatio-Temporal Descriptor Based on 3-D Gradients." BMVC 2008. And the provided poster.



## (iii) Orientation quantization

- ✓ 3D gradients are quantized using a regular n-sided polyhedron
- ✓ The center point of each face corresponds to a histogram bin
- ✓ Efficient quantization via projection of gradient vector on bin axes
- ✓ We use *dodecahedron* (12 sides) and *icosahedrons* (20 sides)

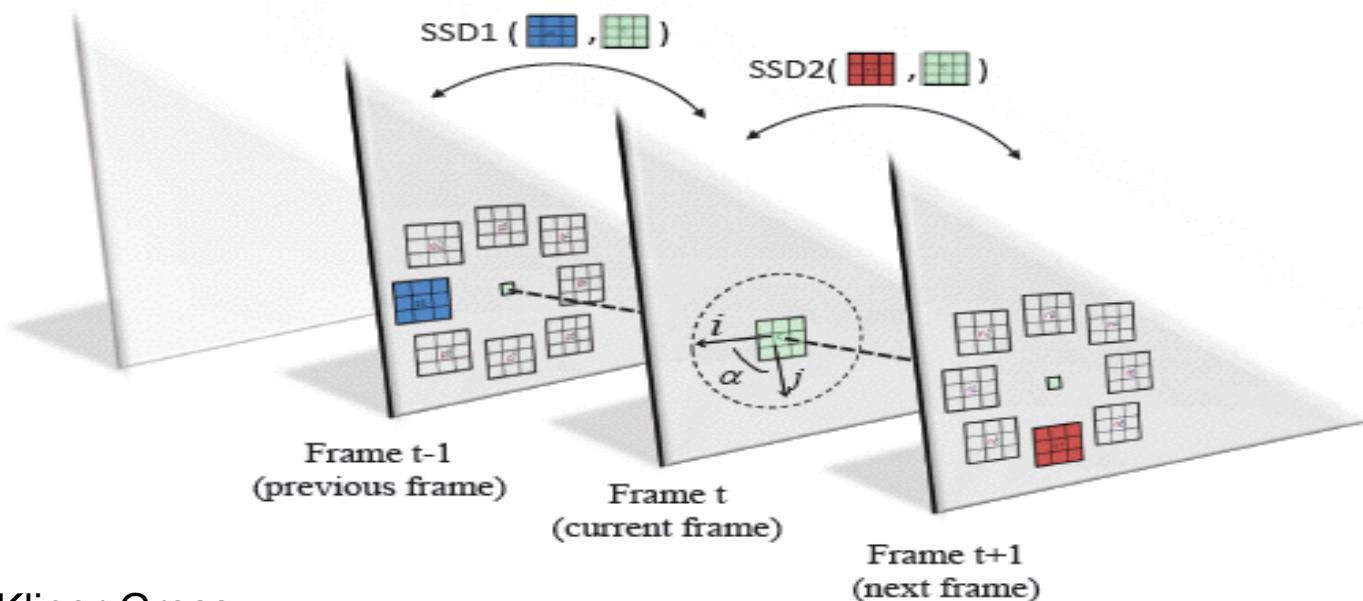
## (iv) Gradient computation

- ✓ Gradients need to be computed at different temporal and spatial scales
- ✓ Other works use a fixed set of pre-computed spatio-temporal scales
- ✓ We propose integral videos
- ✓ Mean gradients can be computed for any spatio-temporal scale

# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

- Local-binary patterns based video descriptor.
  - Dense characterization of motion changes.
  - Captures the shape of moving edges.
  - Methodology incorporates a stabilization mechanism.
- Incorporates a per-pixel encoding using binary/trinary digits.
- Descriptor is frequency of binary/trinary strings.



# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



t

# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



$t-1$



$t$



$t+1$

# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



$t-1$



$t$



$t+1$

$\times$

# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



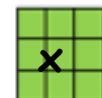
$t-1$



$t$

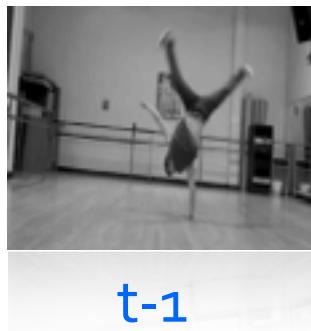


$t+1$

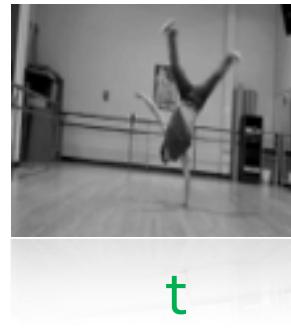


# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



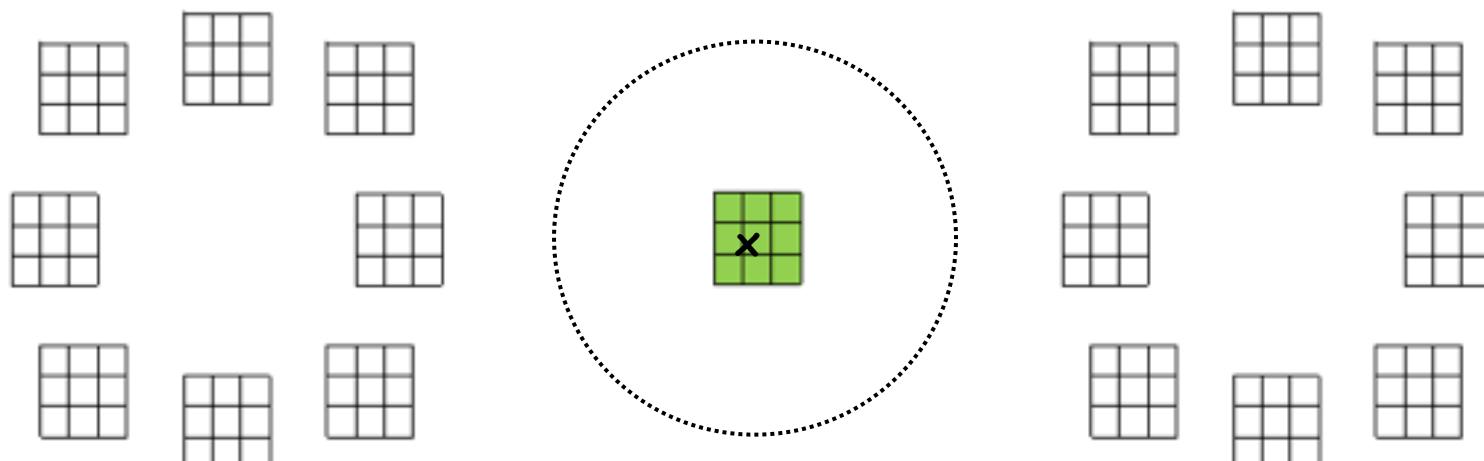
$t-1$



$t$



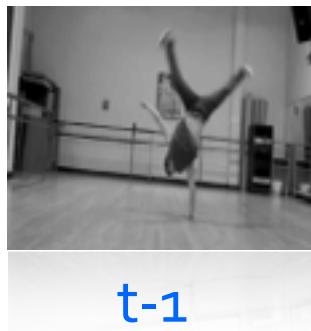
$t+1$



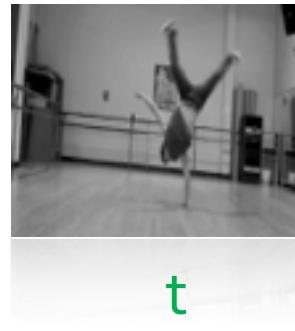
Slide from O.  
Kliper-Gross.

# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



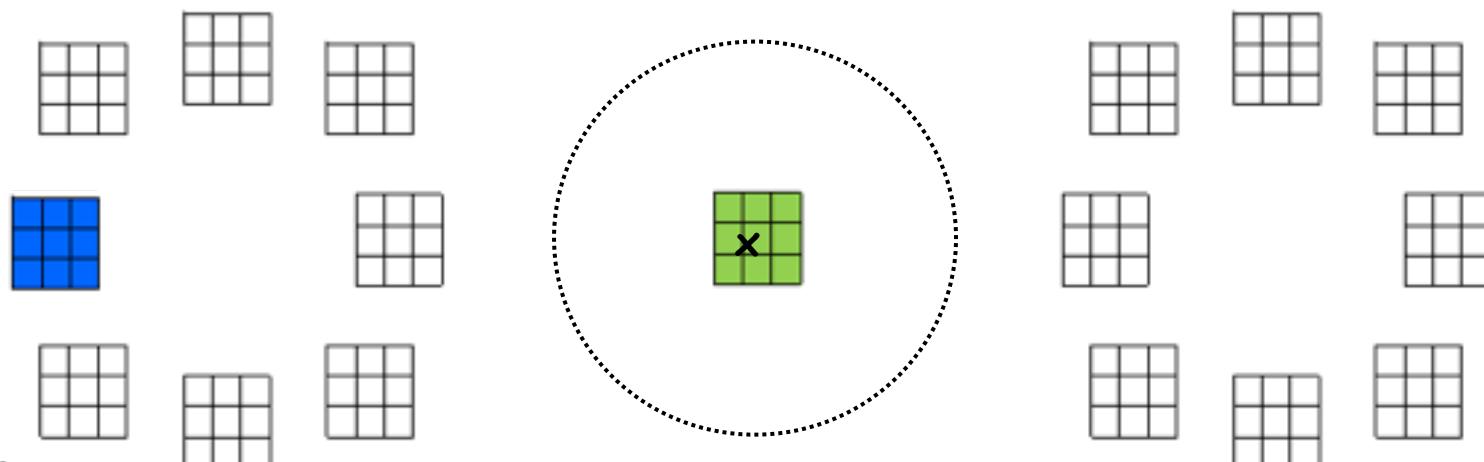
$t-1$



$t$



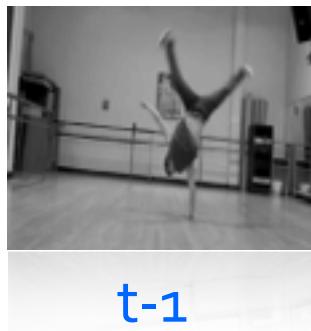
$t+1$



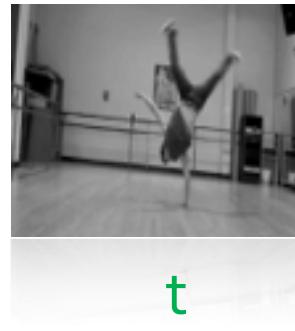
Slide from O.  
Klipper-Gross.

# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



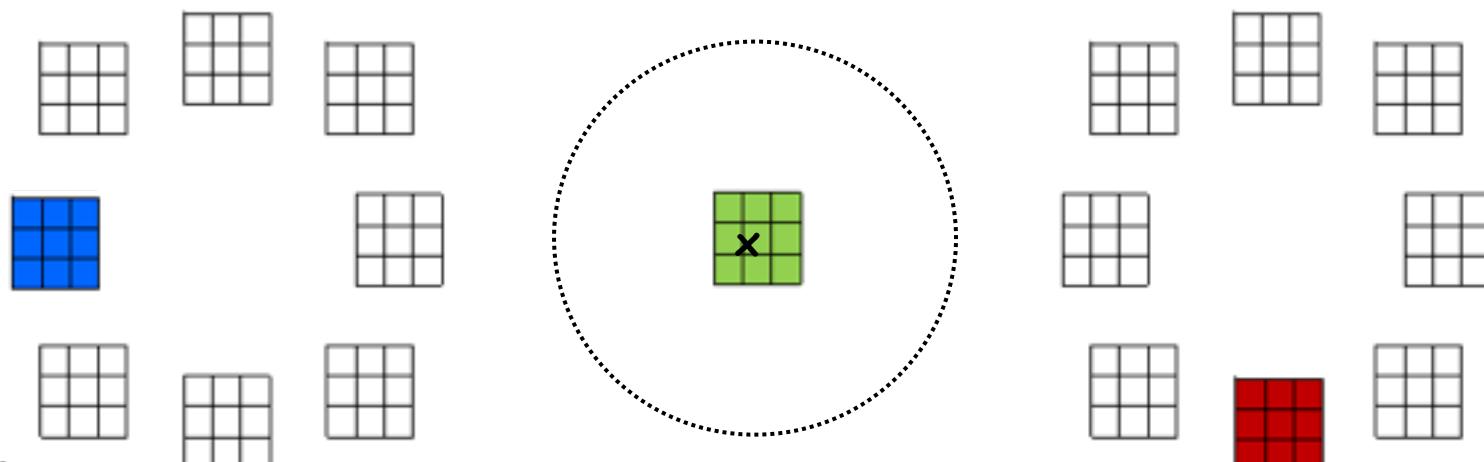
$t-1$



$t$



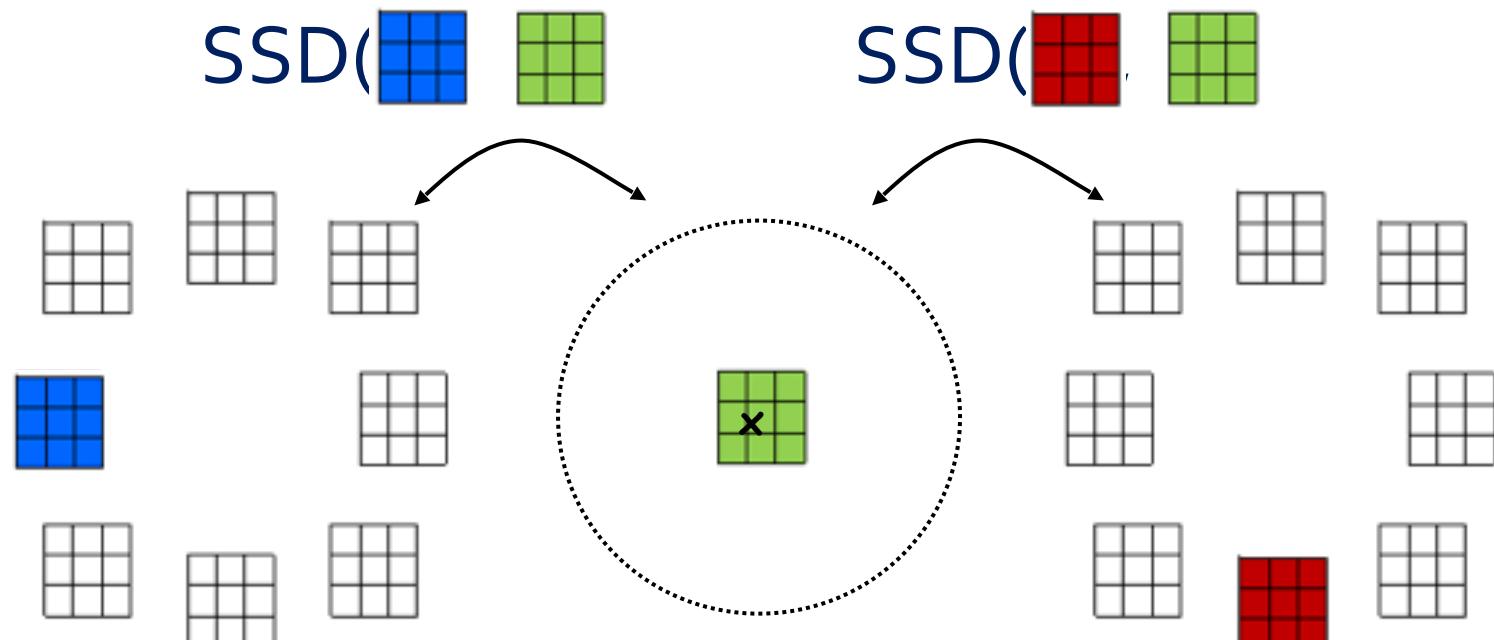
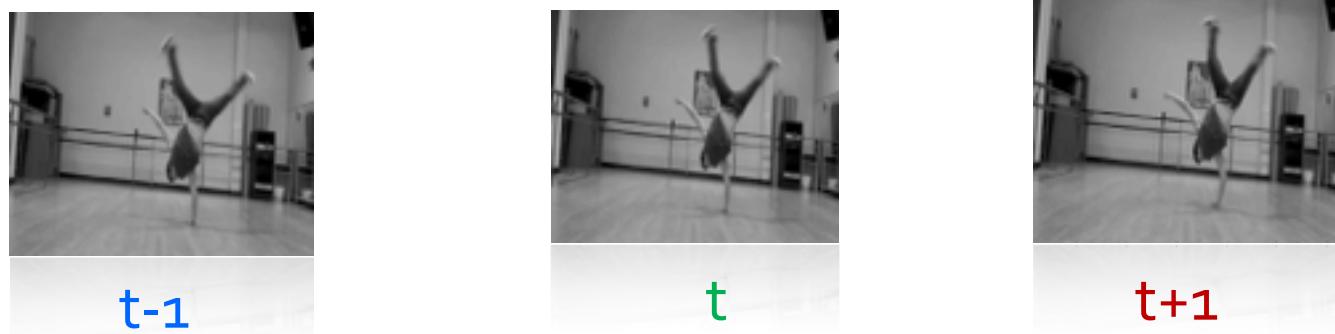
$t+1$



Slide from O.  
Klipper-Gross.

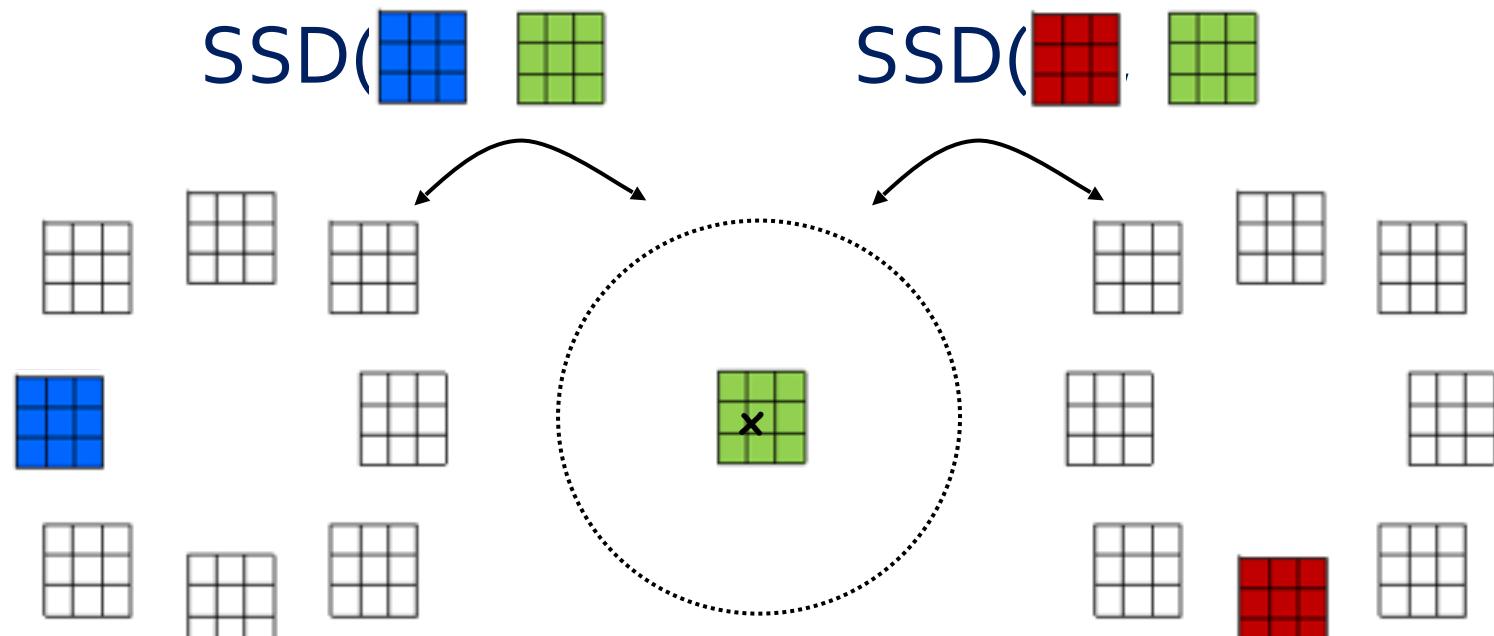
# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



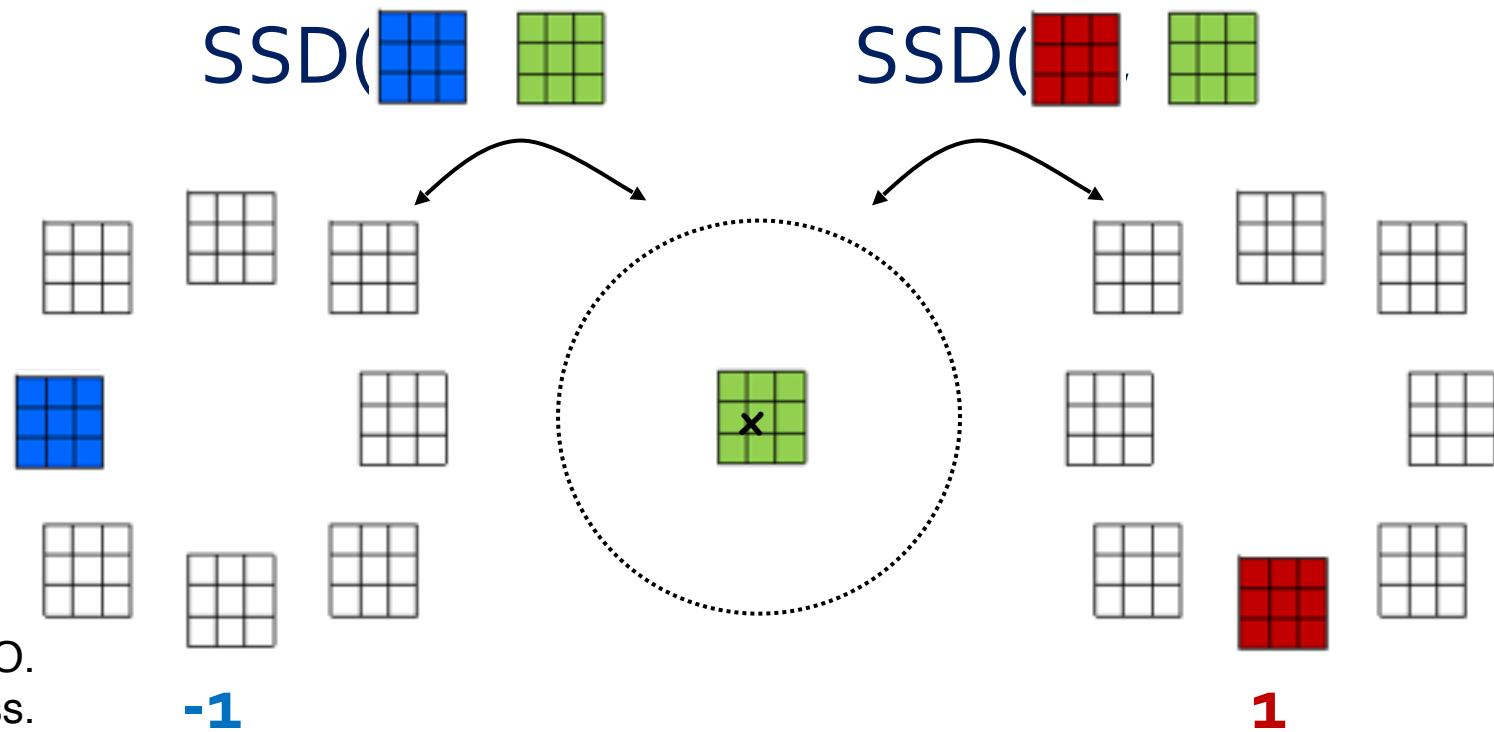
# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



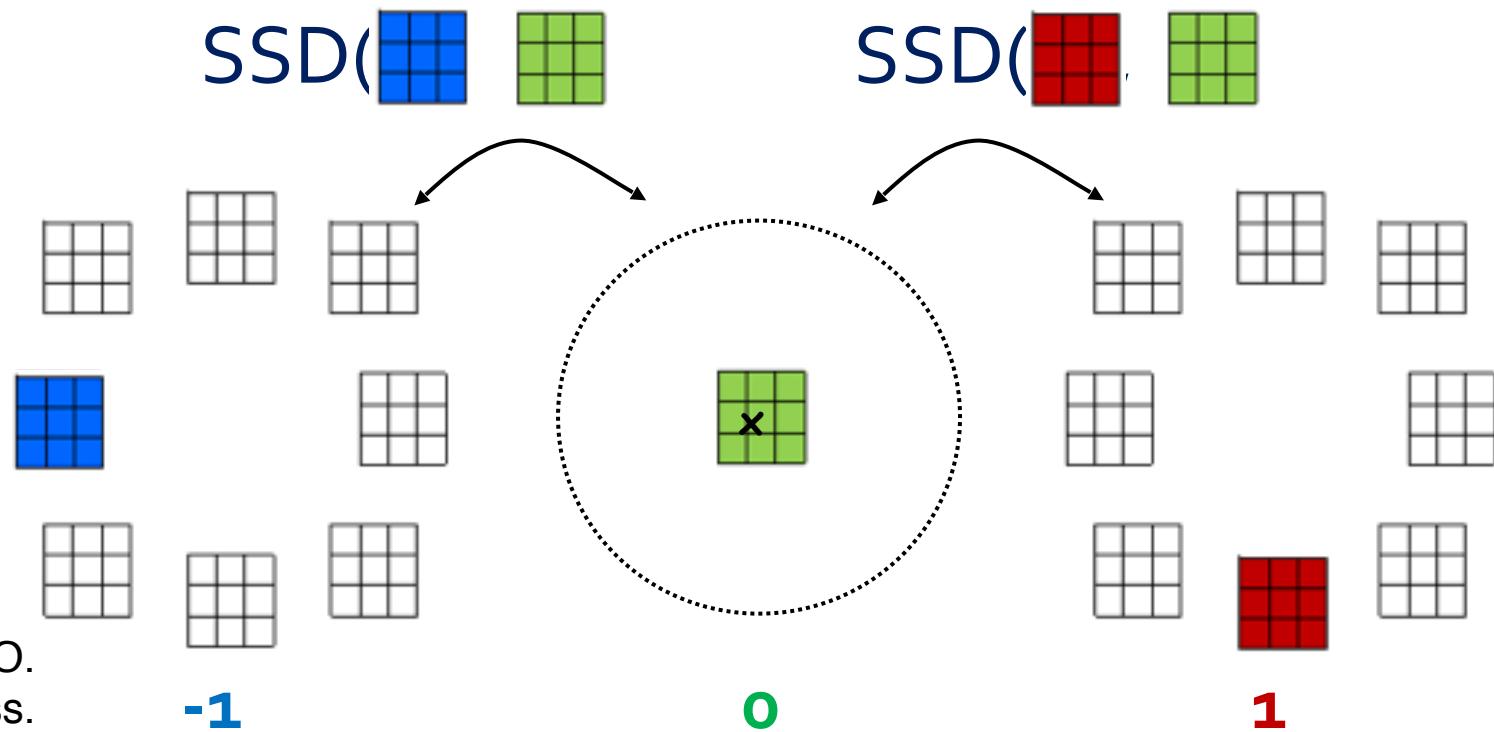
# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



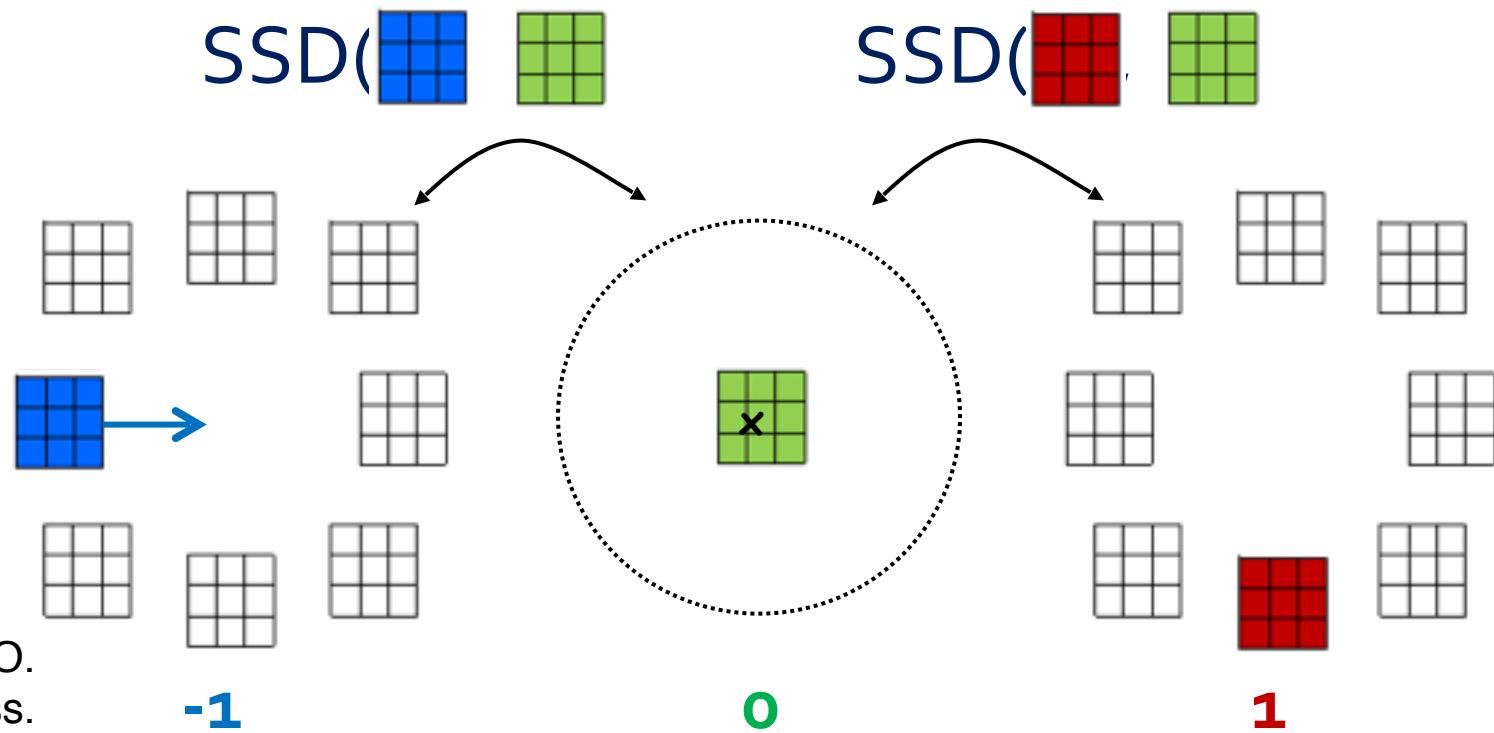
# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



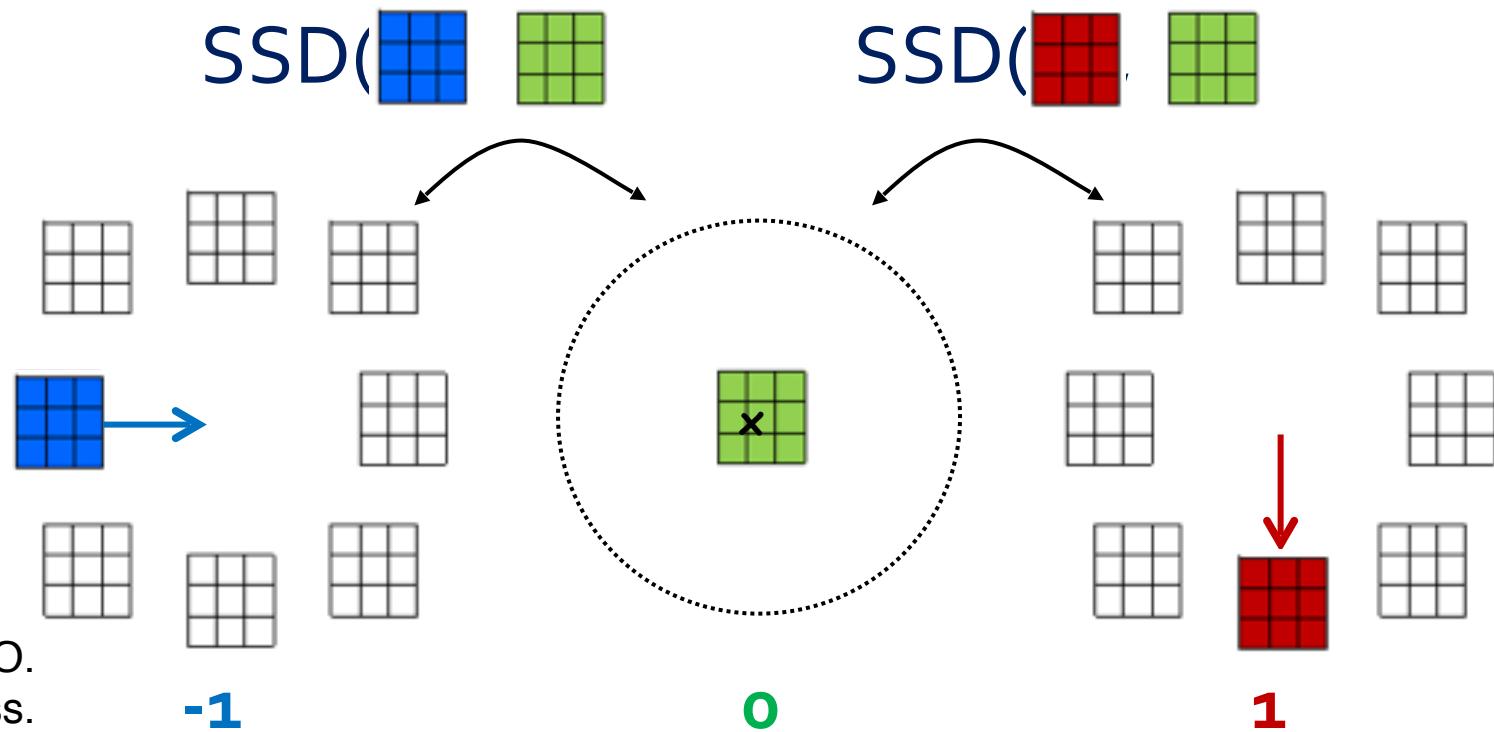
# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



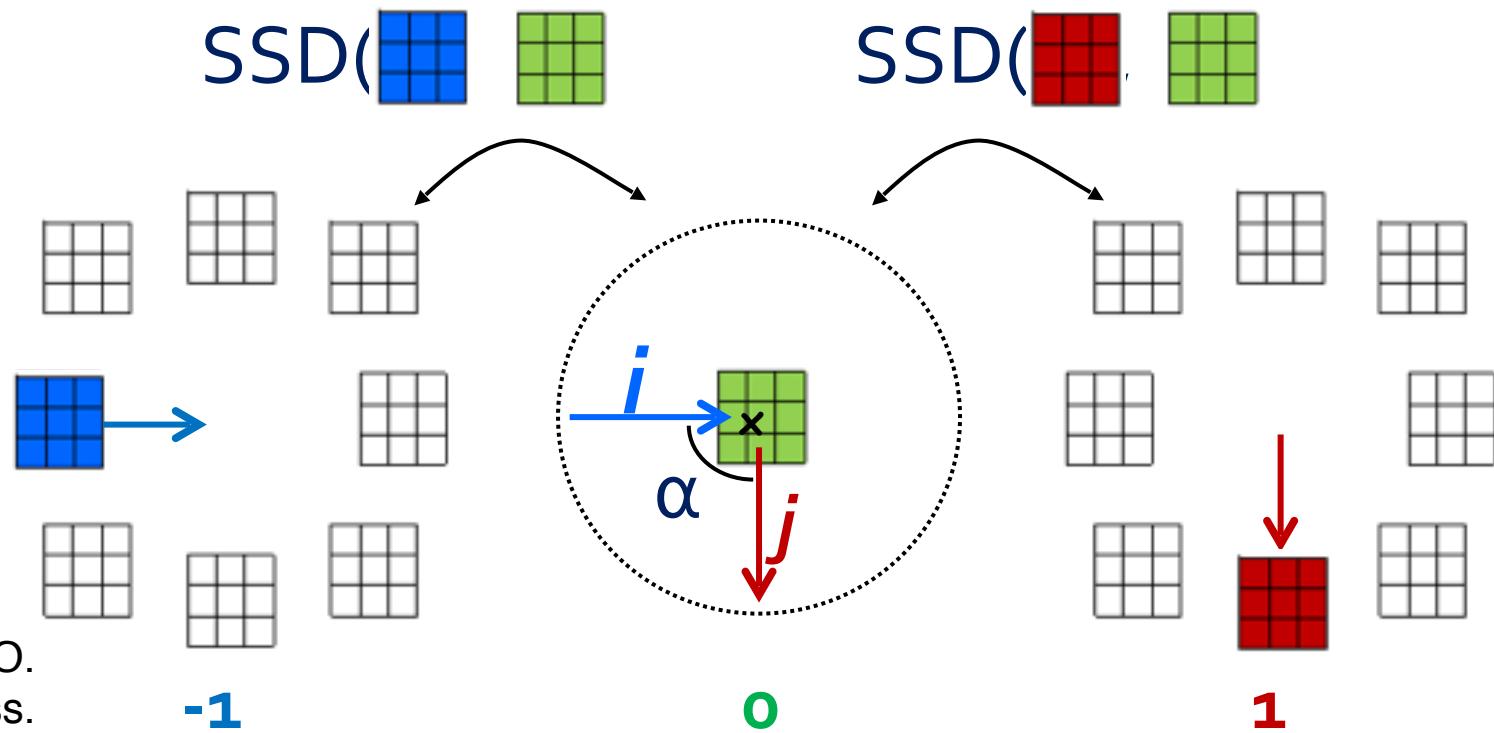
# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



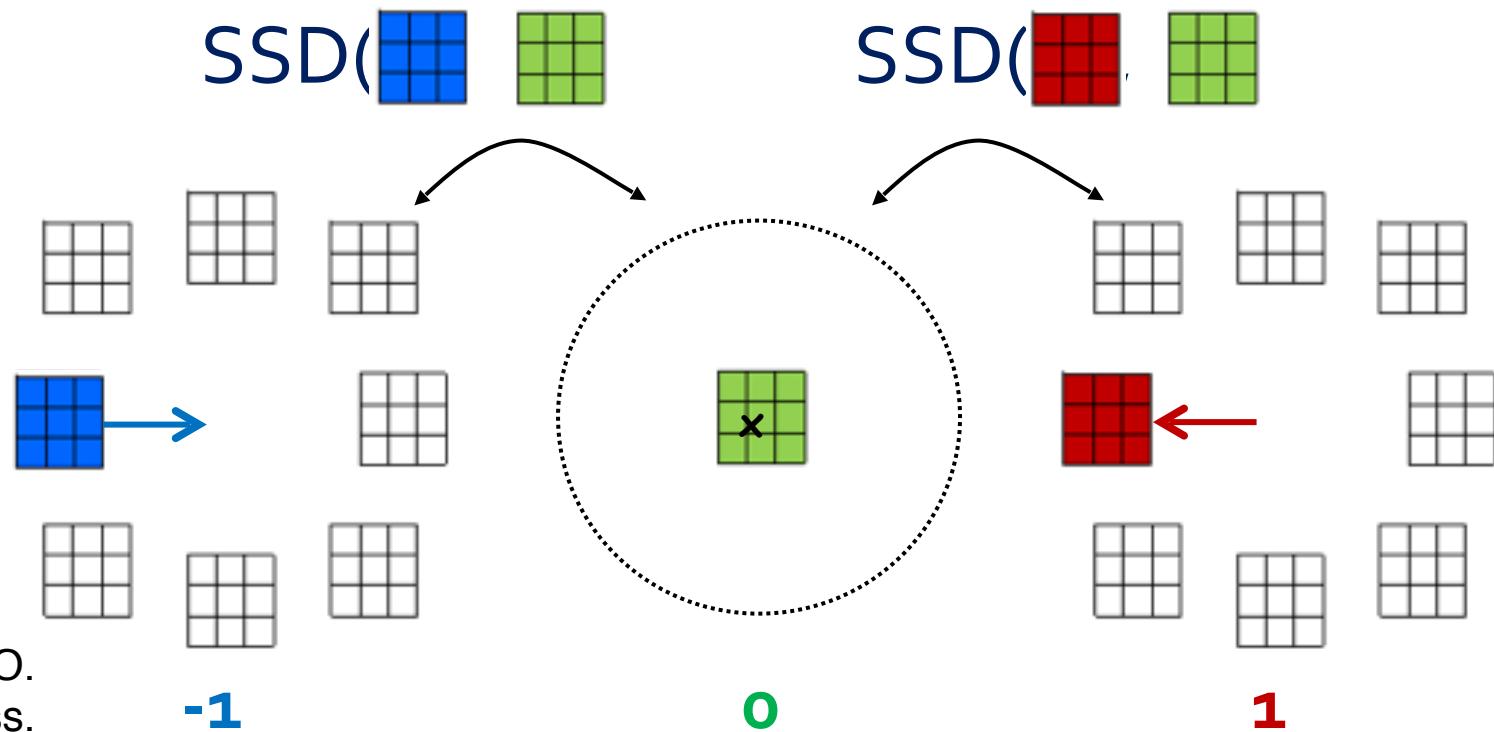
# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



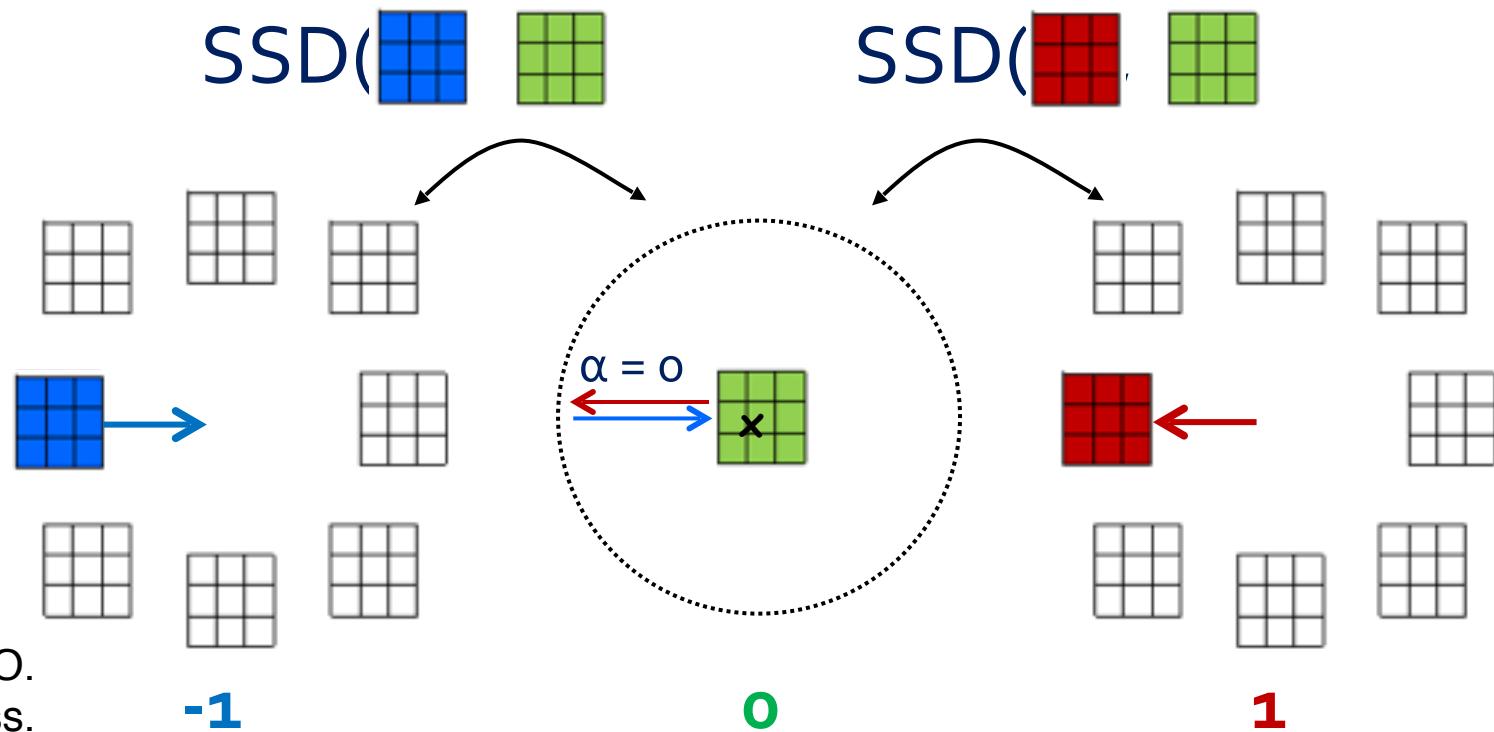
# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



# Local Descriptors: Motion Interchange Patterns

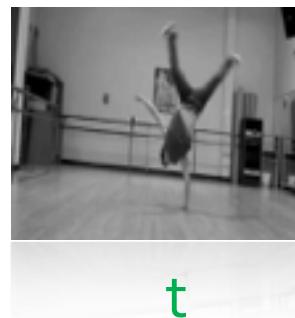
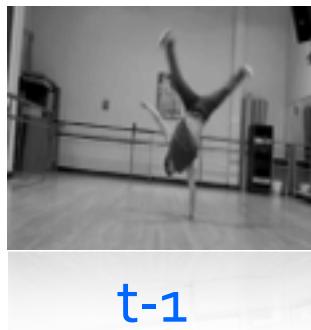
Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



# Local Descriptors: Motion Interchange Patterns

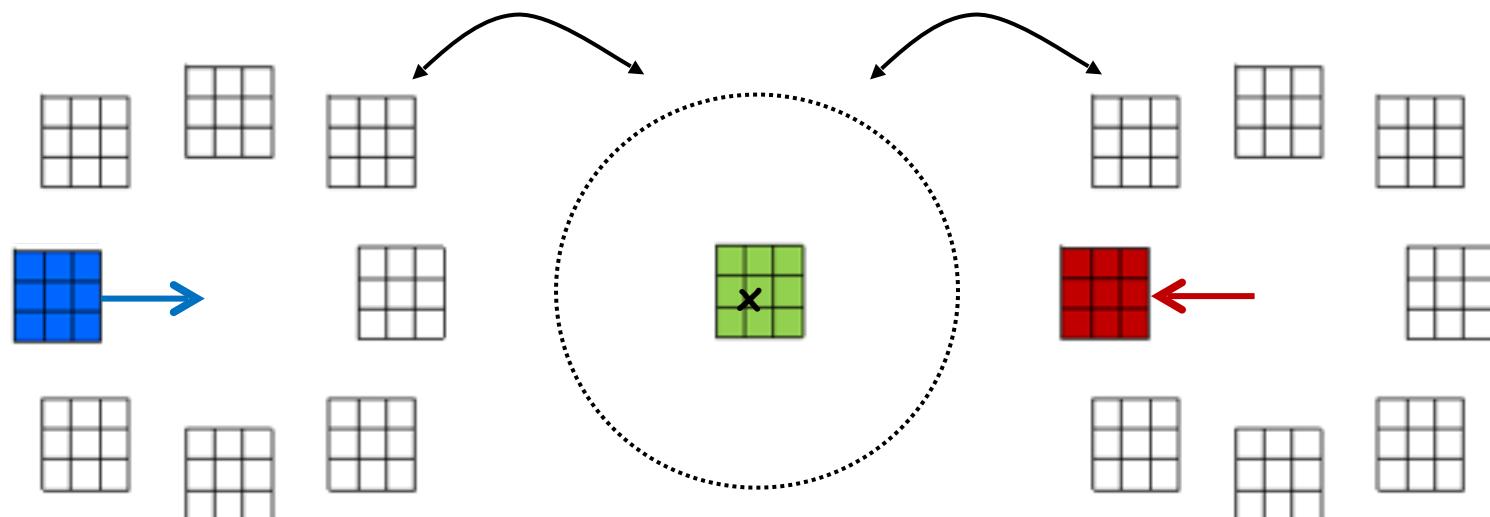
Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

$$\alpha = 0$$



$$SSD(\text{blue grid}, \text{green grid})$$

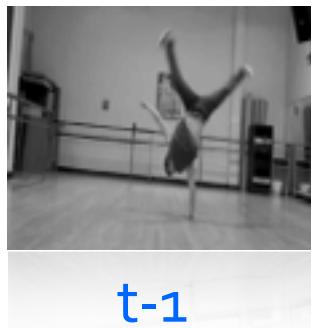
$$SSD(\text{red grid}, \text{green grid})$$



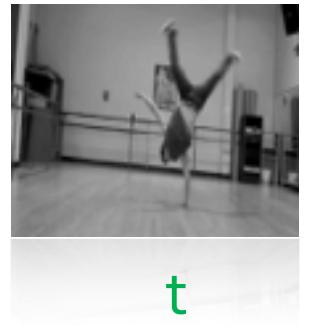
# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

$$\alpha = 0$$



$t-1$



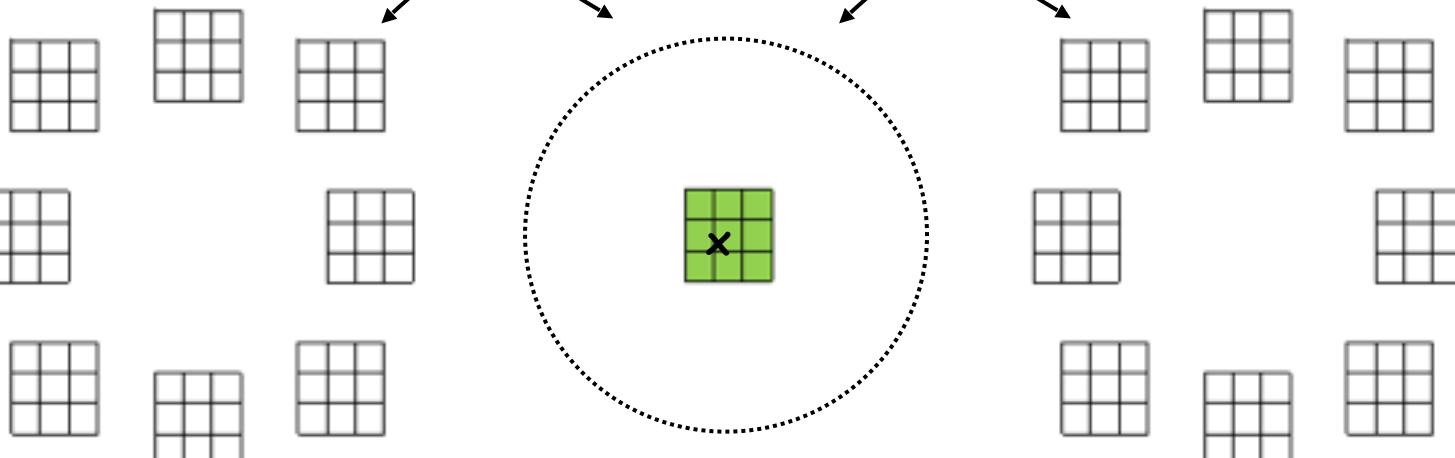
$t$



$t+1$

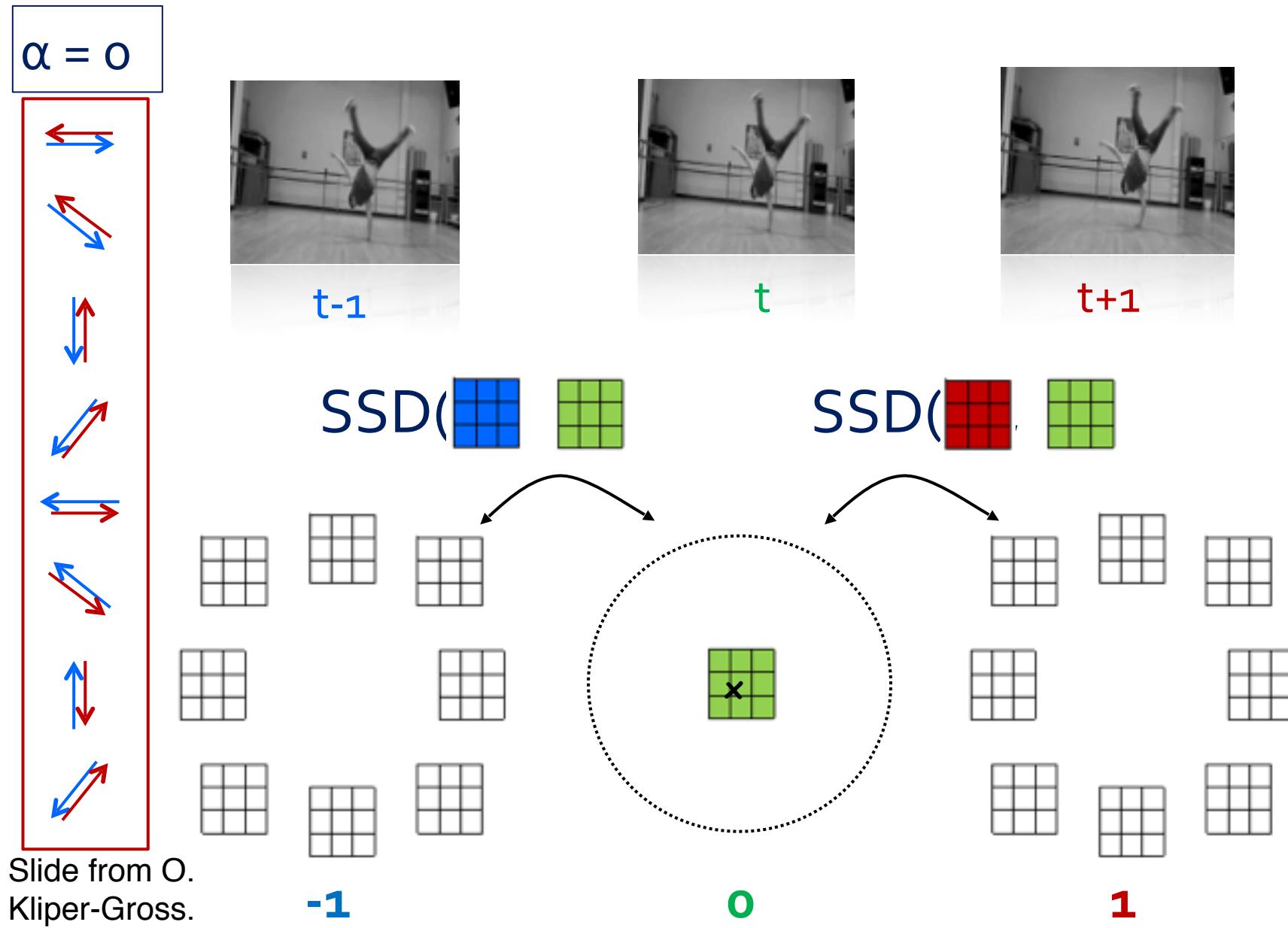
$SSD(\text{blue grid}, \text{green grid})$

$SSD(\text{red grid}, \text{green grid})$



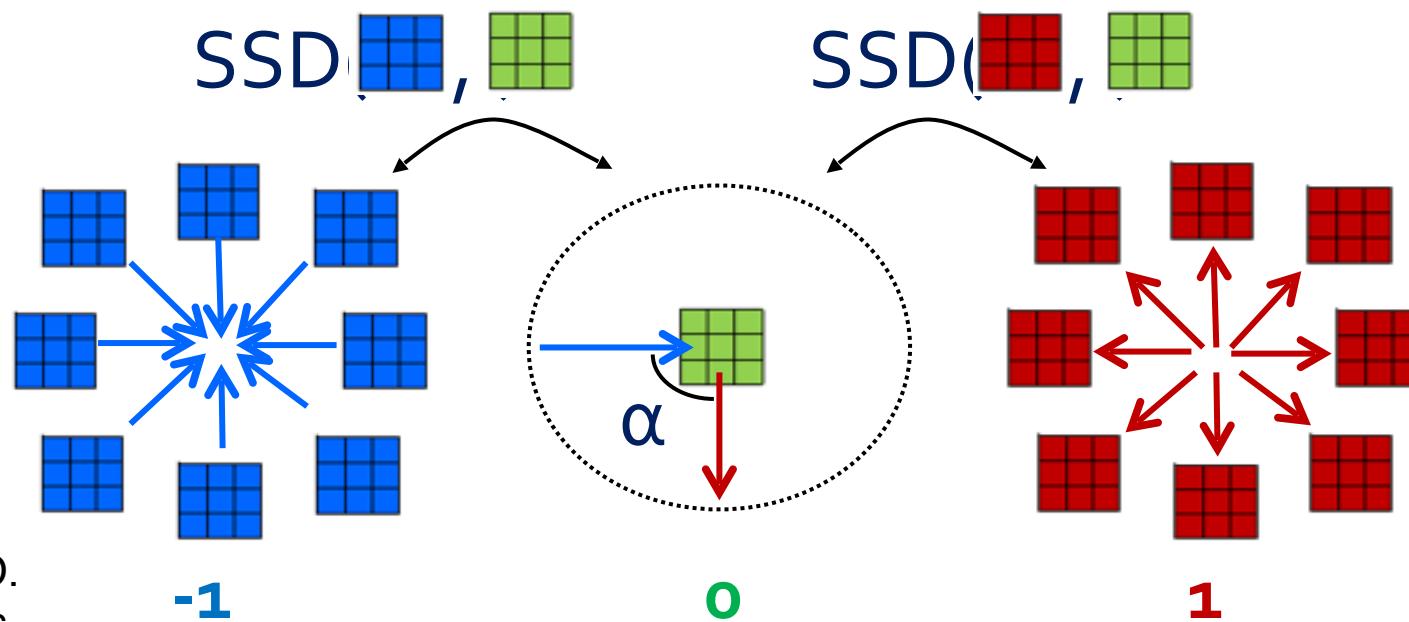
# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



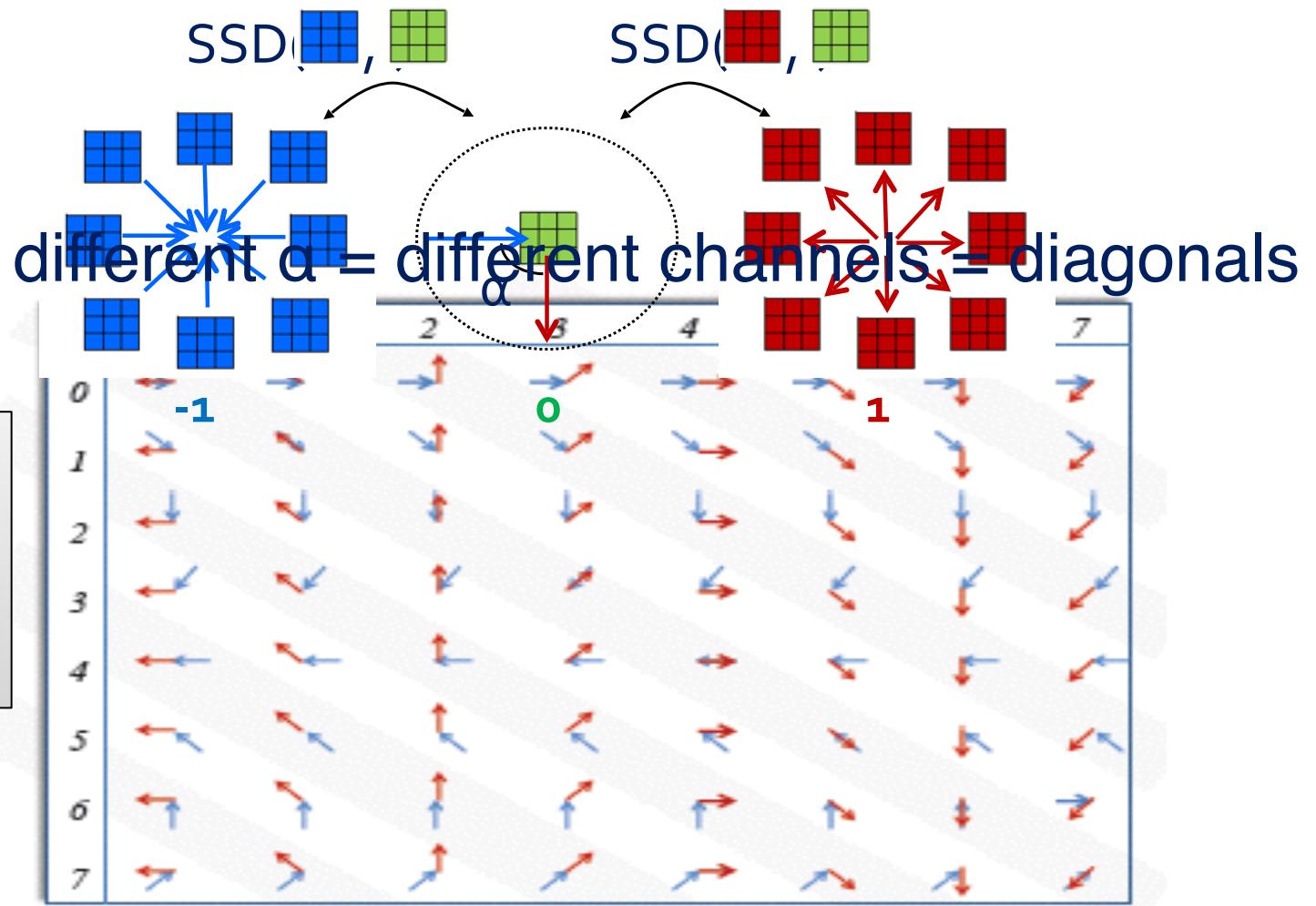
# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

*Per-pixel 64-digits  
trinary code*

# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

Each a defines a channel → *8 channels*

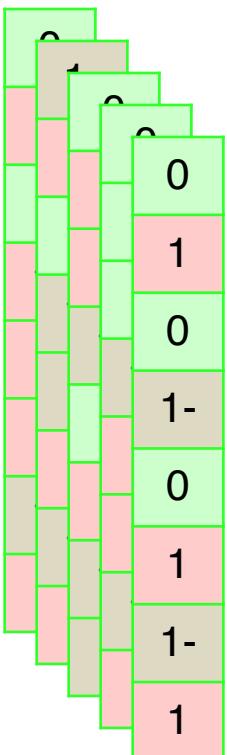
*Per-pixel 64-digits  
trinary code*

# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

Each  $a$  defines a channel → *8 channels*

*Per-pixel 64-digits  
trinary code*

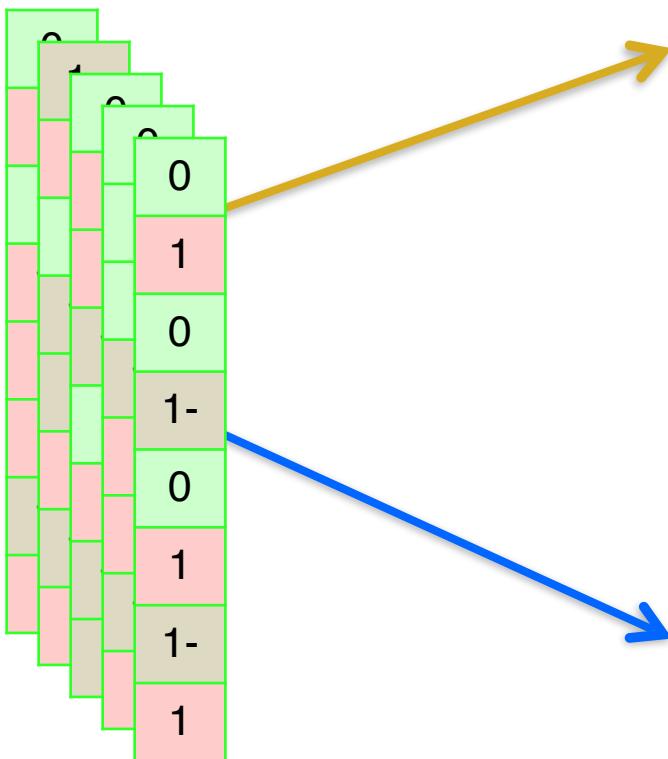


# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

Each  $a$  defines a channel → *8 channels*

*Per-pixel 64-digits  
trinary code*



0	1	0	0	0	1	0	1
0	1	0	0	0	1	0	1
0	1	0	0	0	1	0	1
0	1	0	0	0	1	0	1
0	1	0	0	0	1	0	1
0	1	0	0	0	1	0	1
0	1	0	0	0	1	0	1
0	1	0	0	0	1	0	1



0-255 integer

2 integers per-pixel  
*Per Channel*



0-255 integer

# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



Slide from O.  
Kliper-Gross.

# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

## An example - one channel basic coding

- Vote for next frame
- Vote for prev frame
- Static edges



# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

## An example - one channel basic coding

- Vote for next frame
- Vote for prev frame
- Static edges



Slide from O.  
Kliper-Gross.

MIP captures:  
Motion, Motion Changes, and Shape

# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

# Local Descriptors: Motion Interchange Patterns

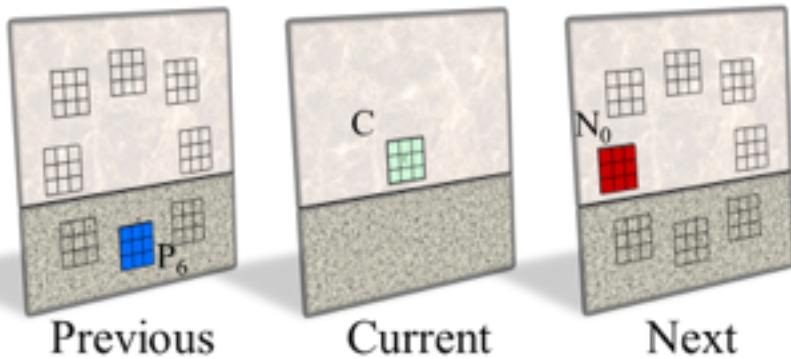
Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

Suppress background structure and noise

# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

Suppress background structure and noise

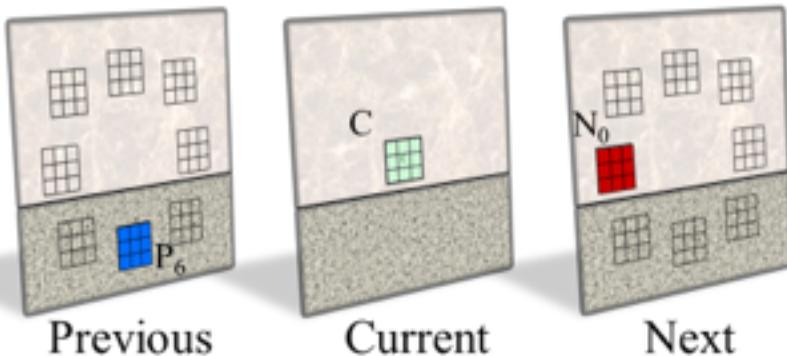


# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

## Suppress background structure and noise

Original Coding = 1

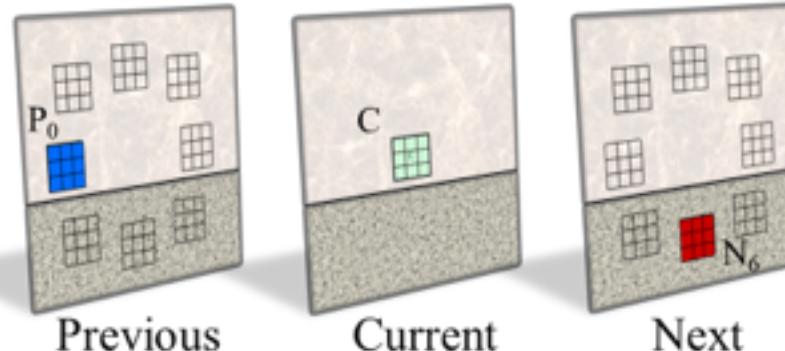
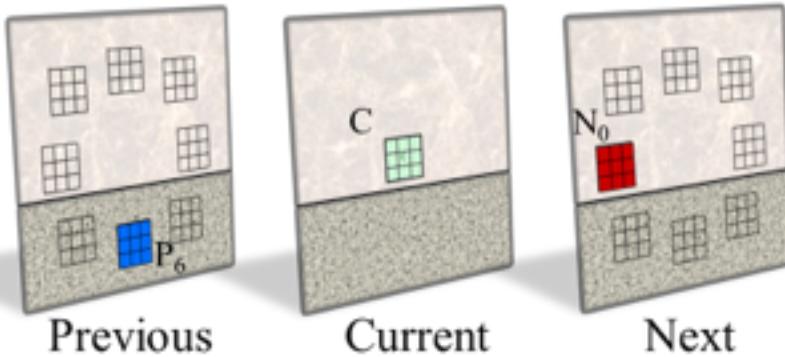


# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

## Suppress background structure and noise

Original Coding = 1

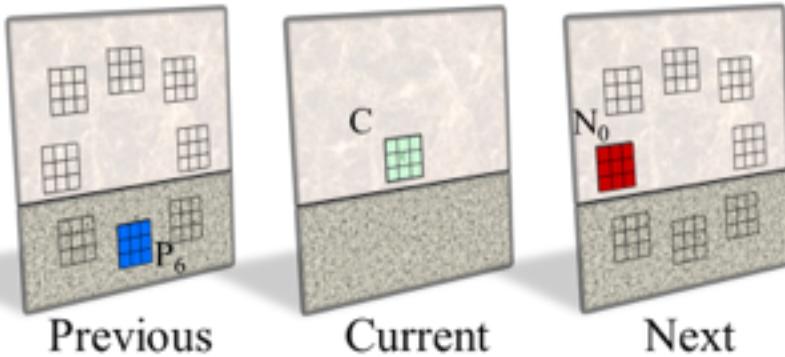


# Local Descriptors: Motion Interchange Patterns

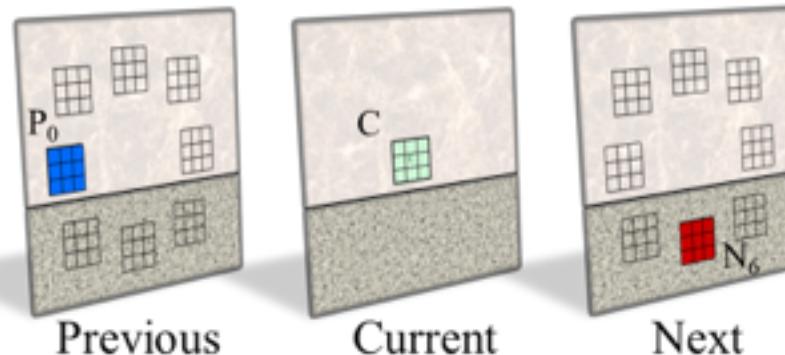
Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

## Suppress background structure and noise

Original Coding = **1**



Switched Locations Coding = **-1**

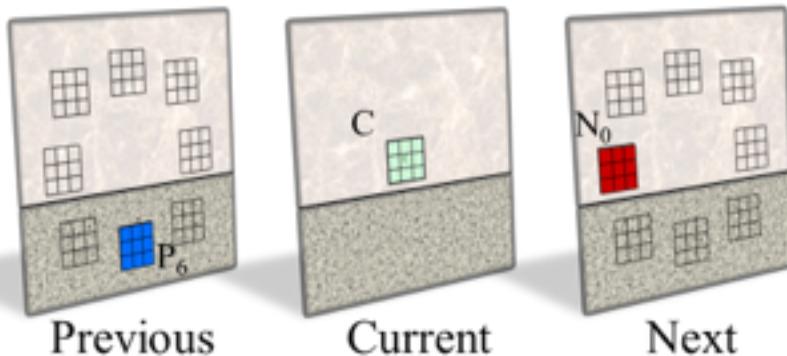


# Local Descriptors: Motion Interchange Patterns

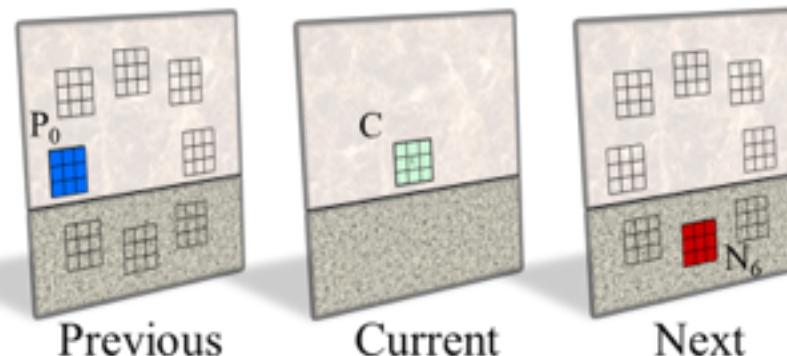
Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

## Suppress background structure and noise

Original Coding = **1**



Switched Locations Coding = **-1**

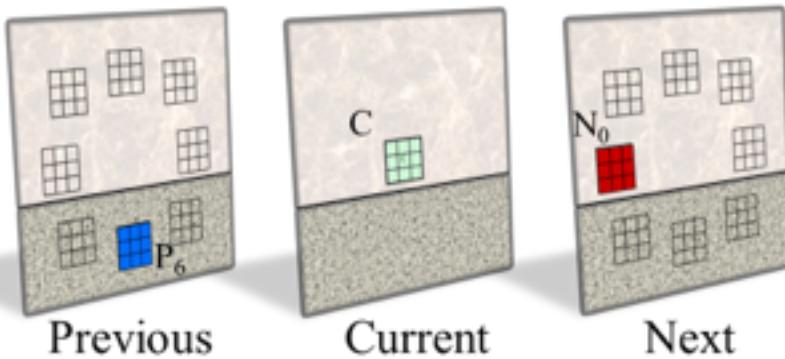


# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

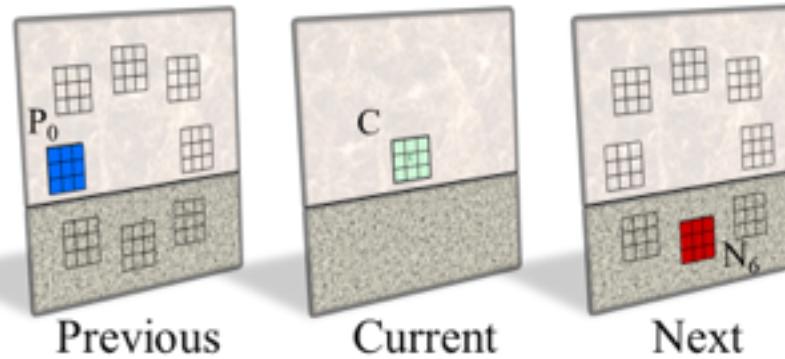
Suppress background structure and noise

Original Coding = 1



2 ways to look at this:

Switched Locations Coding = -1

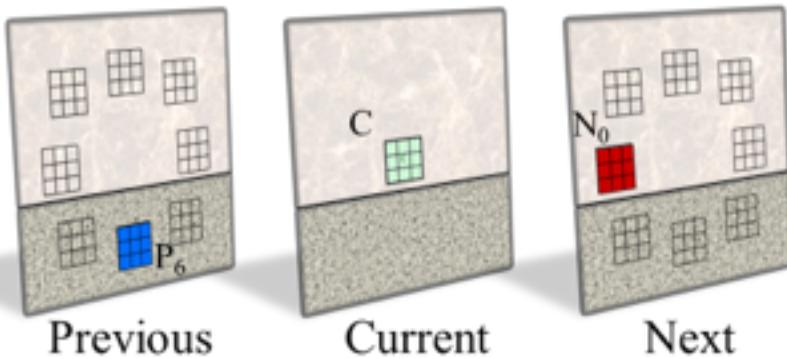


# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

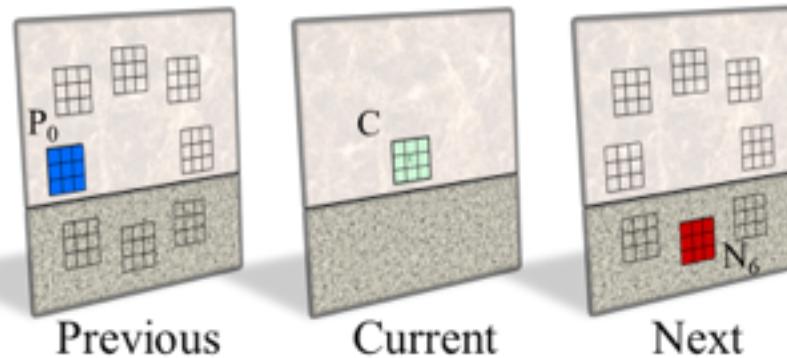
## Suppress background structure and noise

Original Coding =  $1$



2 ways to look at this:  
- No motion.

Switched Locations Coding =  $-1$

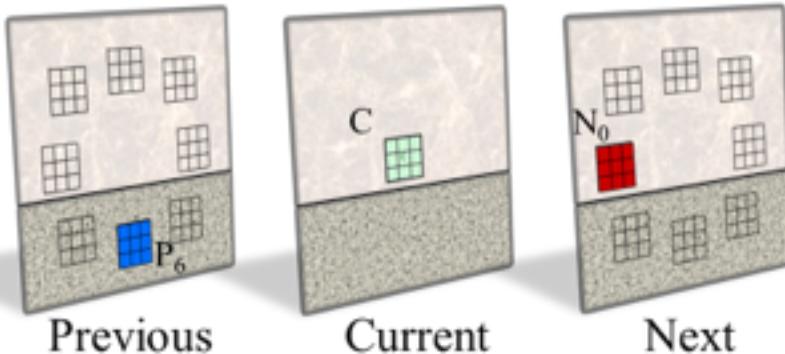


# Local Descriptors: Motion Interchange Patterns

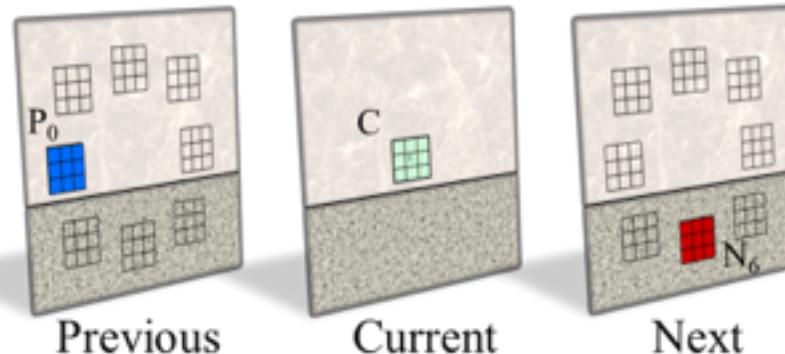
Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

## Suppress background structure and noise

Original Coding =  $1$



Switched Locations Coding =  $-1$



2 ways to look at this:

- No motion.
- Contradicted motion voting.  
i.e.

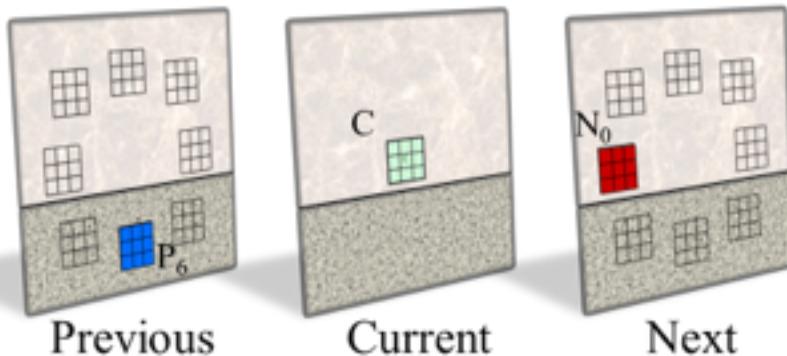
Original coding voted down  $\leftarrow$   
Switched patches voted up  $\rightarrow$

# Local Descriptors: Motion Interchange Patterns

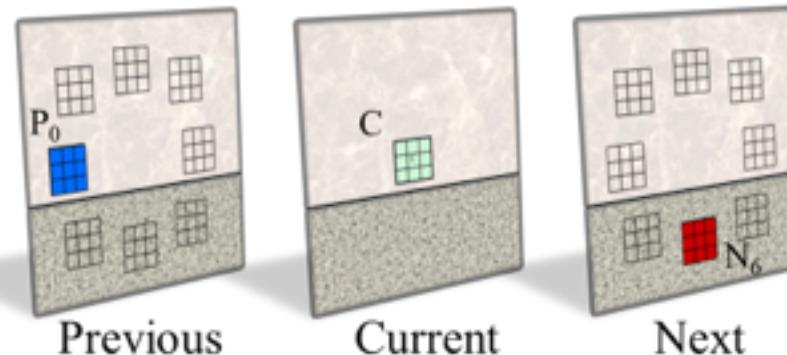
Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

## Suppress background structure and noise

Original Coding =  $1$



Switched Locations Coding =  $-1$



### Switched Patch Suppression

2 ways to look at this:

- No motion.
- Contradicted motion voting.  
i.e.

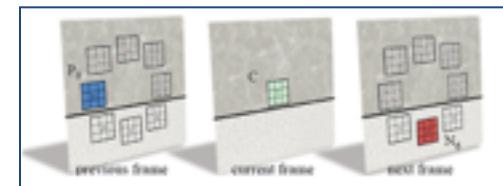
Original coding voted down  $\leftarrow$   
Switched patches voted up  $\rightarrow$



Suppress the code

# Local Descriptors: Motion Interchange Patterns

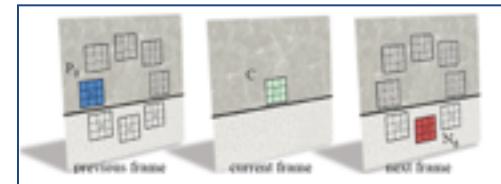
Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



# Local Descriptors: Motion Interchange Patterns

Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

An example of MIP suppression.



Without  
Suppression

Original

With  
Suppression

