COLLEGE OF ENGINEERING
ELECTRICAL ENGINEERING & COMPUTER SCIENCE
UNIVERSITY OF MICHIGAN

Case Study for Geometric Invariance Local Image Features

EECS 504 Computer Vision

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

Live In Class Starts on Slide 26

Materials on these slides have come from many sources in addition to myself; specific sources are cited on each slide.

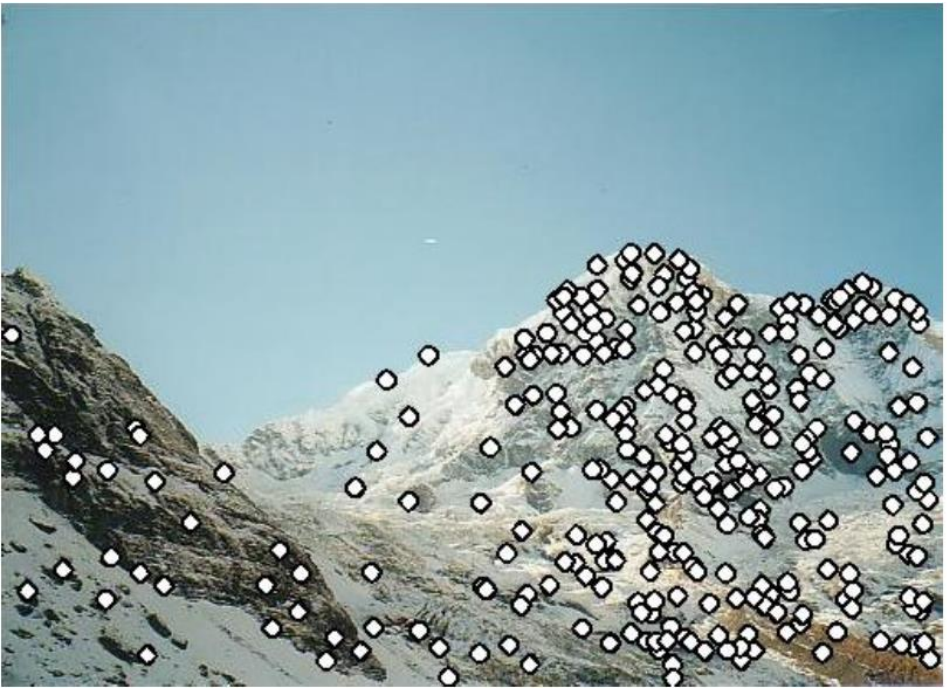
Plan

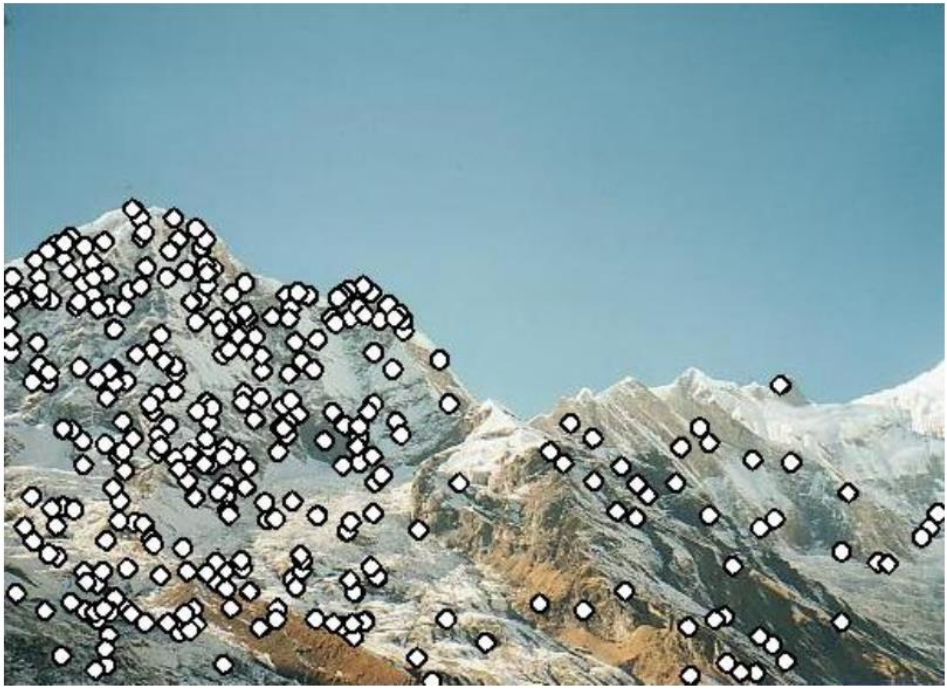
- What are local image features and why are they useful.
- Local Image Feature Detection
- Invariance
- Local Image Feature Description

Consider an Application: Image Stitching

Sources for this example: I. Kokkinos, D. Frolova, D. Simakov.

Consider an Application: Image Stitching

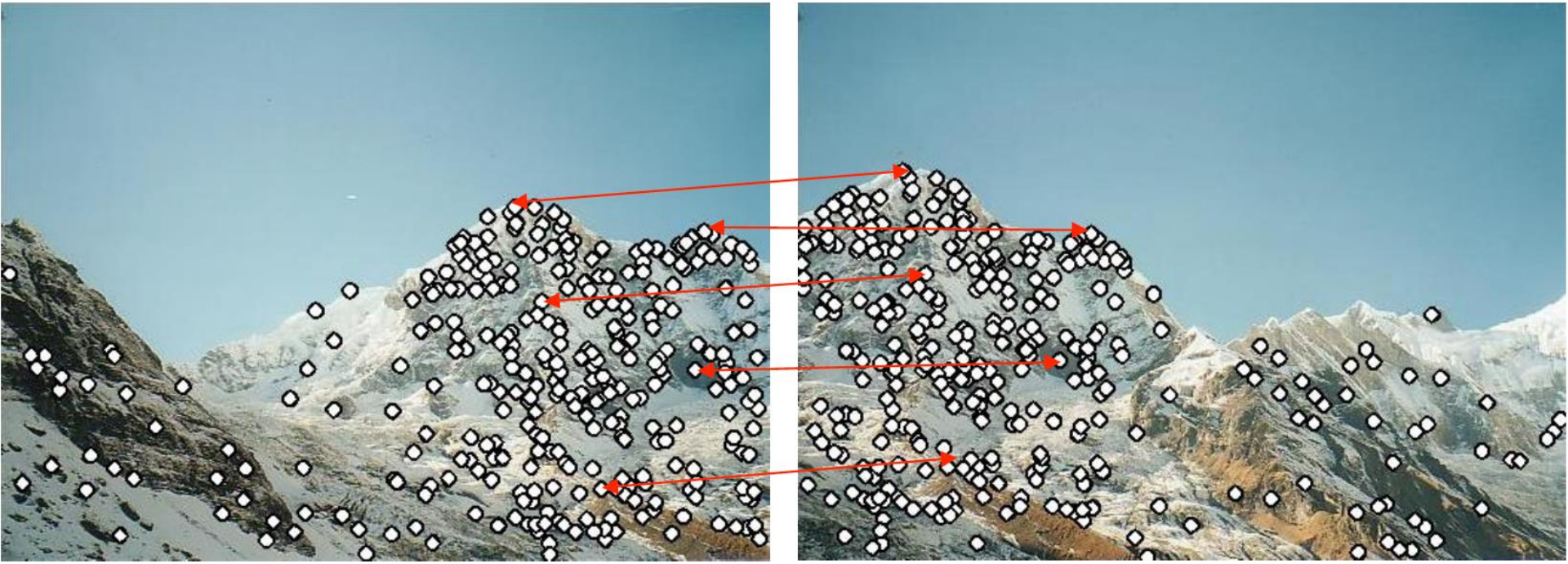




1. Detect feature points in both images.

Sources for this example: I. Kokkinos, D. Frolova, D. Simakov.

Consider an Application: Image Stitching



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

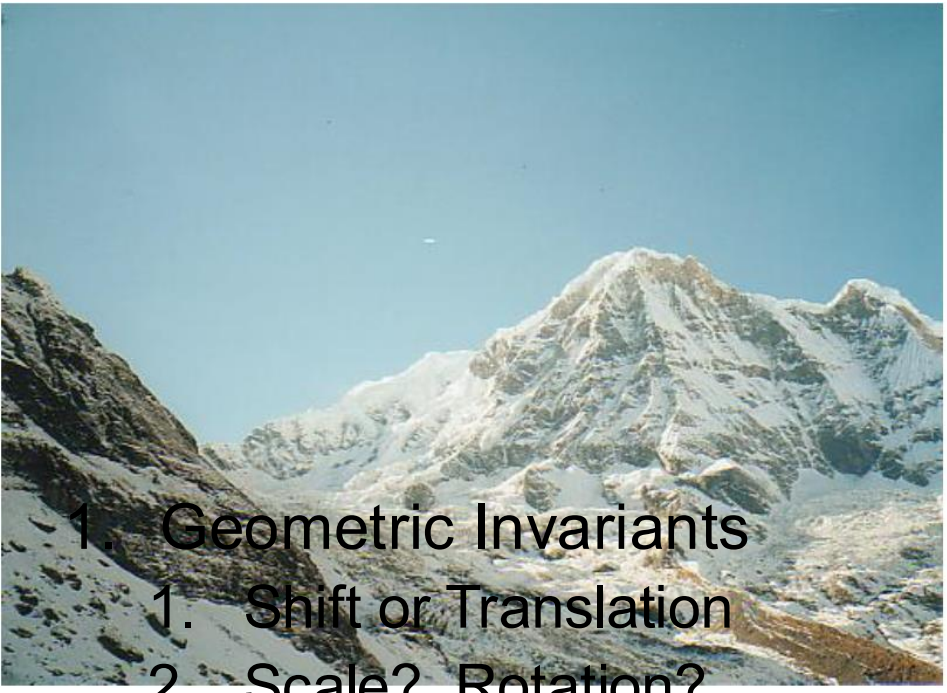
Sources for this example: I. Kokkinos, D. Frolova, D. Simakov.

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**

Sources for this example: I. Kokkinos, D. Frolova, D. Simakov.

What invariants do we care about here?



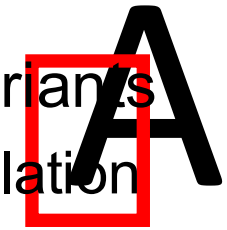
1. Geometric Invariants
    1. Shift or Translation
    2. Scale? Rotation?
    3. Affine?
    4. Viewpoint?
2. Scene layout?
3. Photometric invariants?

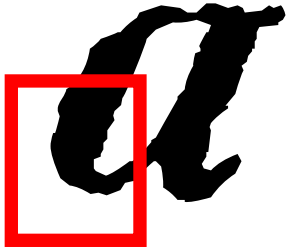
Consider an Application: Detect Object Instances

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

Sources for this example: S. Savarese.

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. **Reduction**
2. Find corresponding pairs of feature points. **Matching**
3. Use the pairs to solve objective function. **Estimation**

- Other applications of local image features
  - 3D reconstruction
  - Motion tracking
  - Object recognition
  - Indexing and database retrieval
  - Robot navigation

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

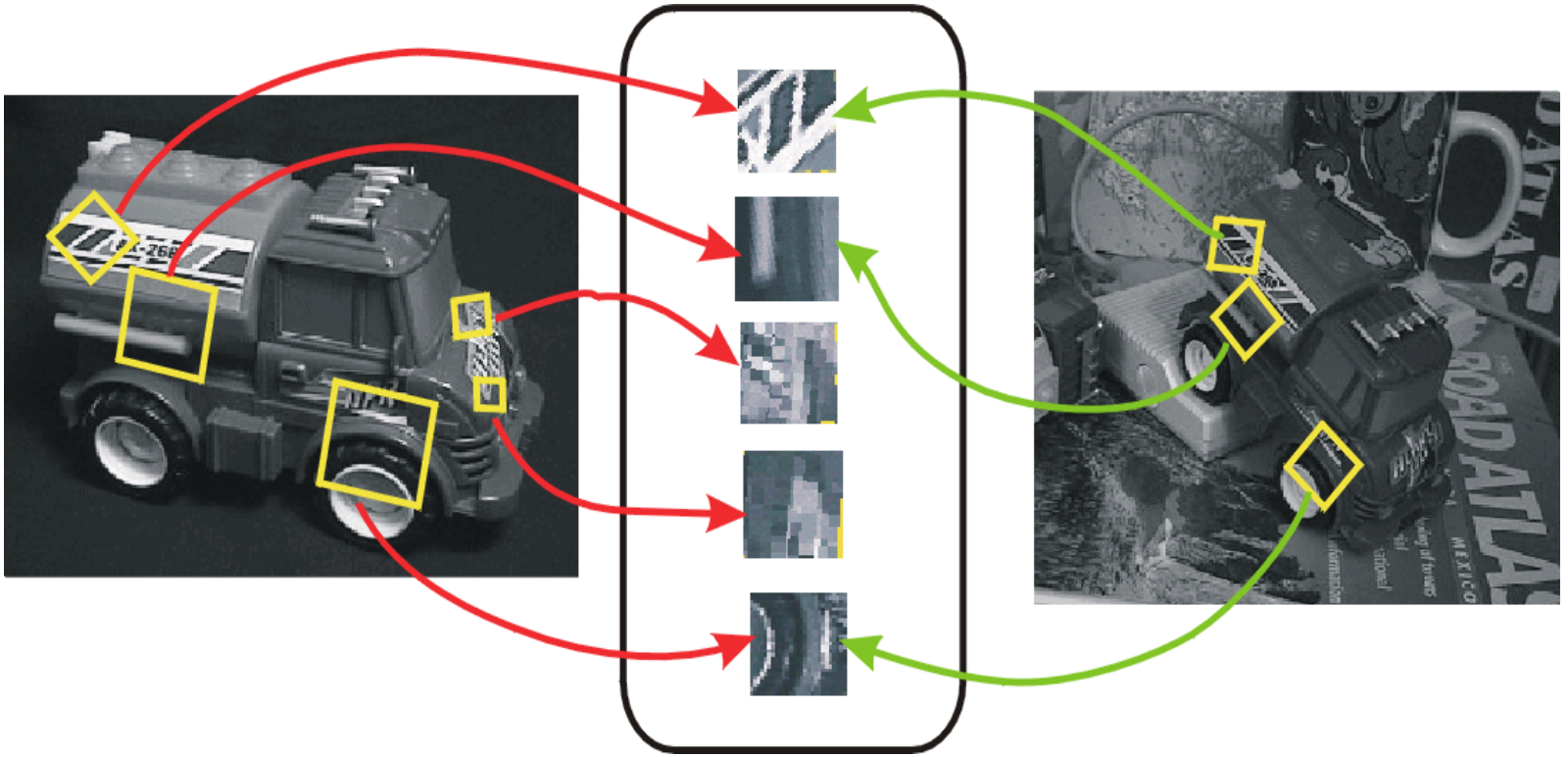
Generality

- exploit different types of features in different situations

Source: S. Seitz slides.

Challenges

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



Sources for this example: I. Kokkinos, D. Frolova, D. Simakov, S. Seitz.

What makes a good feature?

A good

B neutral

C bad

Source for this example: S. Seitz.

Repeatability

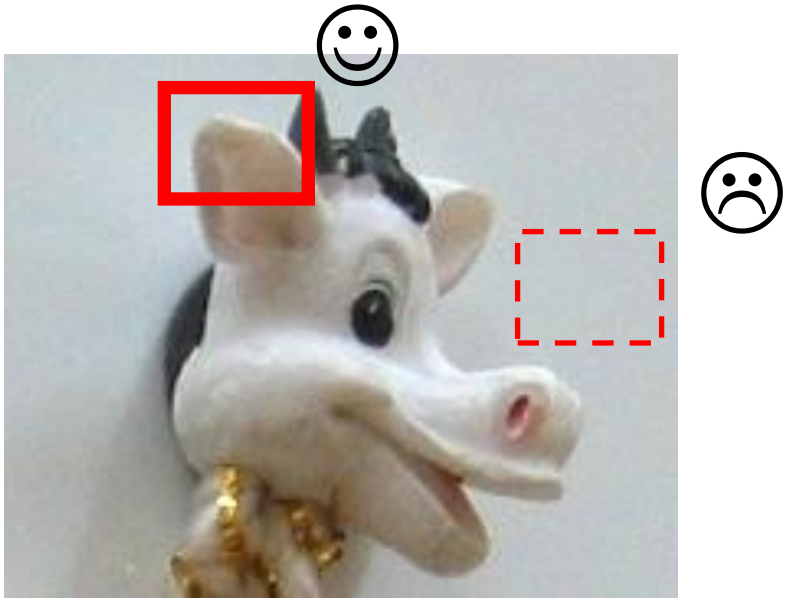
Illumination invariance

Scale invariance

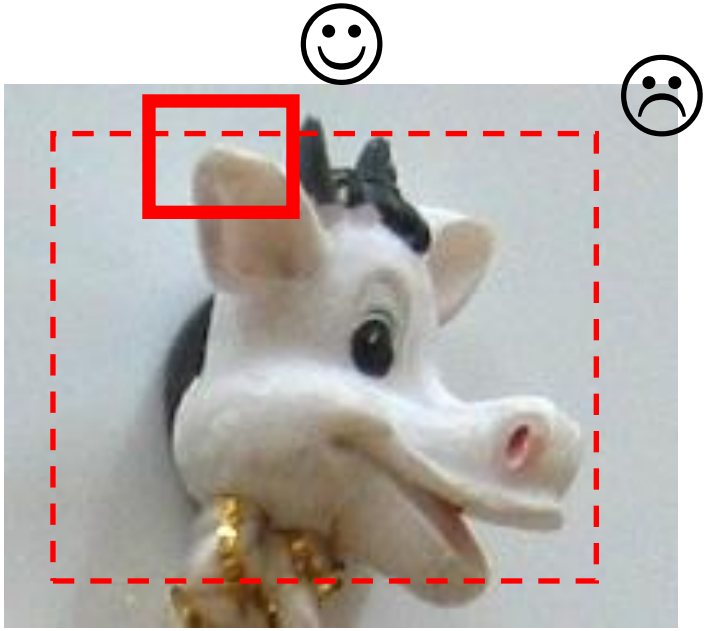
Pose invariance
- Rotation
- Affine

Source for this example: Savarese.

- Saliency



- Locality



Source for this example: Savarese.

One criterion is uniqueness

Look for image regions that are unusual

- Lead to unambiguous matches in other images

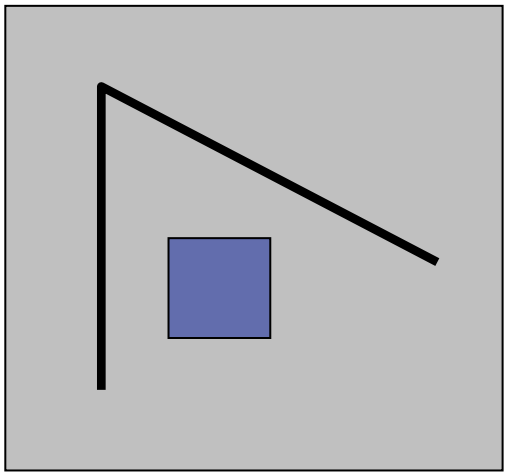
How to define “unusual”?

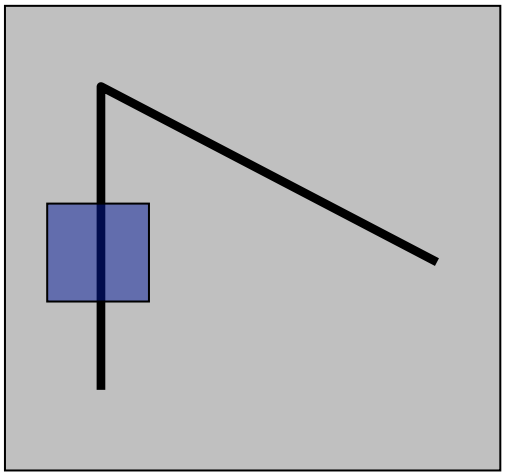
Source: S. Seitz.

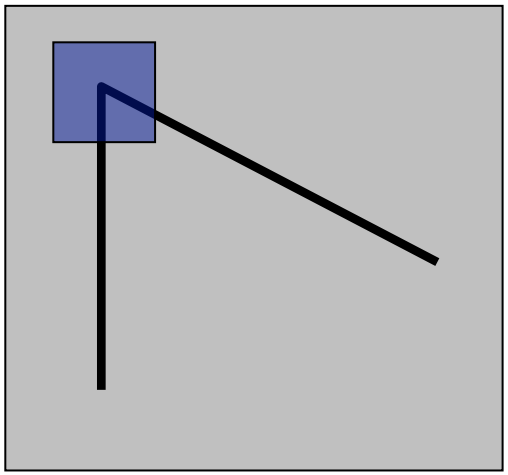
Local measures of uniqueness

Suppose we only consider a small window of pixels

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





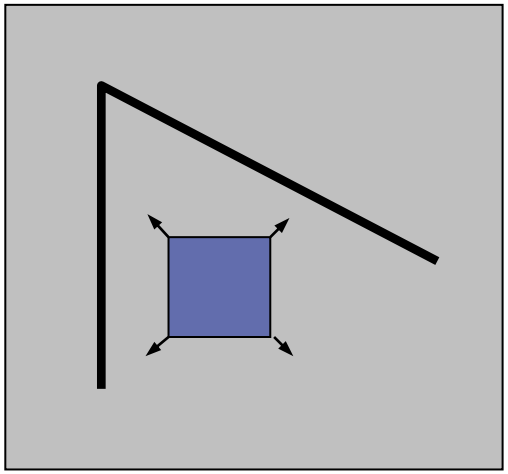


Sources: I. Kokkinos, D. Frolova, D. Simakov, S. Seitz.

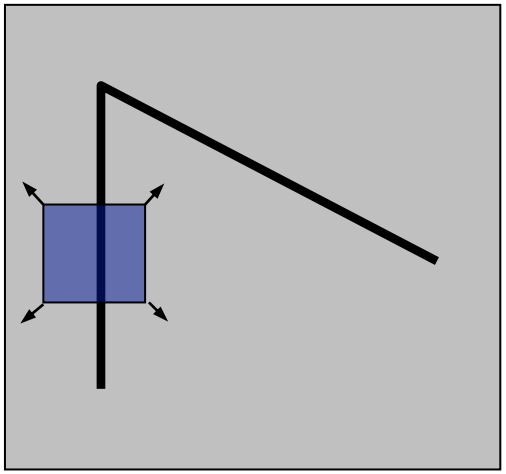
Feature detection

Local measure of feature uniqueness

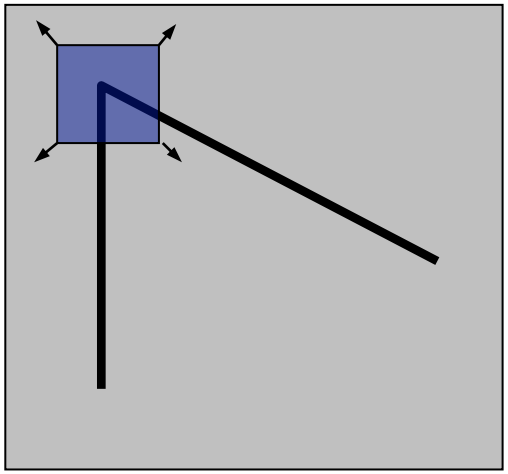
- How does the window change when you shift it?
- Shifting the window in *any direction* causes a *big change*



"flat" region:
no change in all directions



"edge":
no change along the edge direction



"corner":
significant change in all directions

Sources: I. Kokkinos, D. Frolova, D. Simakov, S. Seitz.

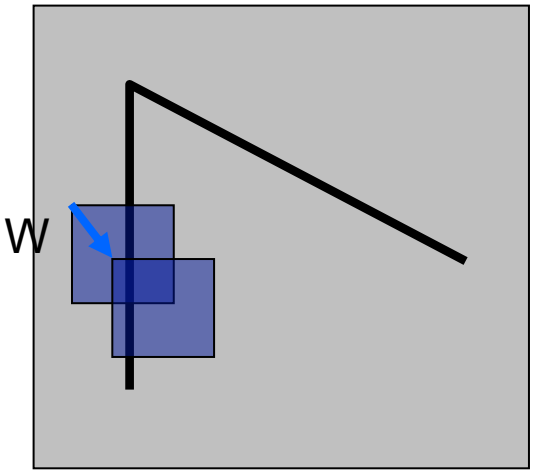
Stop Slides

See PVL for interactive mathematical derivation of the corner operator; slides below capture similar content but with less description

Feature detection: the math

Consider shifting the window W by (u,v)

- how do the pixels in W change?
- compare each pixel before and after by summing up the squared differences (SSD)
- this defines an SSD "error" of *E(u,v)*:



$$E(u,v) = \sum_{(x,y)\in W} [I(x+u, y+v) - I(x,y)]^2$$

Source: S. Seitz.

Small motion assumption

- Taylor Series expansion of I

$$I(x+u, y+v) = I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \text{higher order terms}$$

- If the motion is small, then the first order approx. is good:

$$I(x+u, y+v) \approx I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$$

$$\approx I(x,y) + \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

shorthand

$$I_x = \frac{\partial I}{\partial x}$$

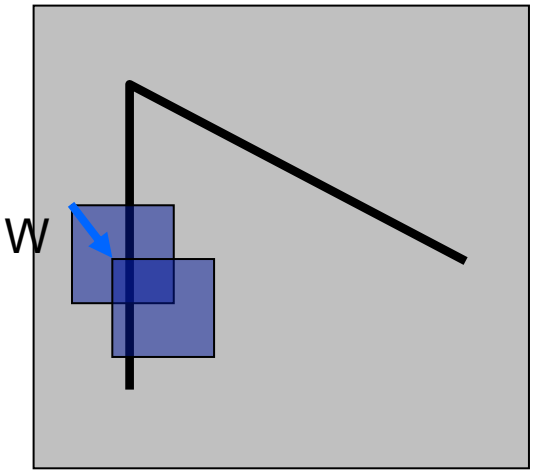
- Plug this back into the objective function.

Source: S. Seitz.

Feature detection: the math

Consider shifting the window W by (u,v)

- how do the pixels in W change?
- compare each pixel before and after by summing up the squared differences
- this defines an "error" of E(u,v):



$$E(u,v) = \sum_{(x,y)\in W} [I(x+u,y+v) - I(x,y)]^2$$

$$\approx \sum_{(x,y)\in W} [I(x,y) + [I_x \; I_y] \begin{bmatrix} u \\ v \end{bmatrix} - I(x,y)]^2$$

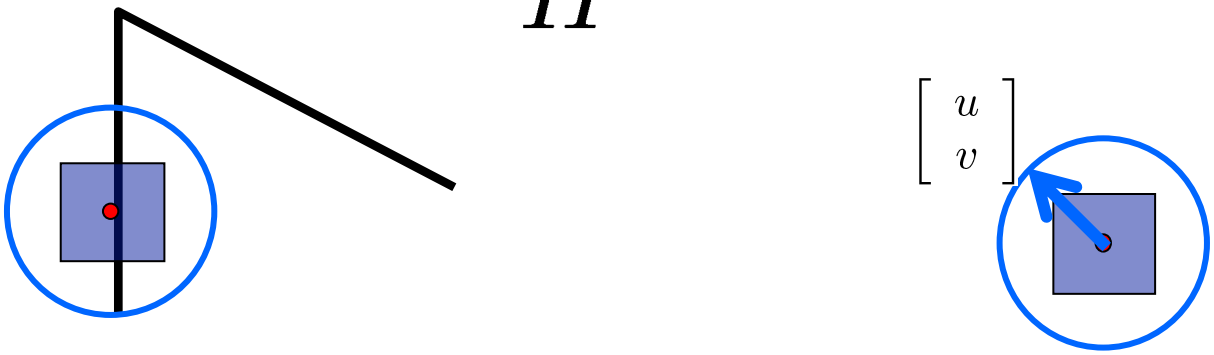
$$\approx \sum_{(x,y)\in W} \left[[I_x \; I_y] \begin{bmatrix} u \\ v \end{bmatrix} \right]^2$$

Source: S. Seitz.

Feature detection: the math

This can be rewritten:

$$E(u,v) = \sum_{(x,y)\in W} [u\ v] \underbrace{\begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}}_{H} \begin{bmatrix} u \\ v \end{bmatrix}$$



For the example above

- You can move the center of the window to anywhere on the blue unit circle
- Which directions will result in the largest and smallest E values?
- We can find these directions by looking at the eigenvectors of *H*

Source: S. Seitz.

Quick eigenvalue/eigenvector review

The **eigenvectors** of a matrix **A** are the vectors **x** that satisfy:

$$Ax = \lambda x$$

The scalar λ is the **eigenvalue** corresponding to **x**

- The eigenvalues are found by solving:

$$det(A - \lambda I) = 0$$

- In our case, ***A*** = ***H*** is a 2x2 matrix, so we have

$$det \begin{bmatrix} h_{11} - \lambda & h_{12} \\ h_{21} & h_{22} - \lambda \end{bmatrix} = 0$$

- The solution:

$$\lambda_{\pm} = \frac{1}{2} \left[(h_{11} + h_{22}) \pm \sqrt{4h_{12}h_{21} + (h_{11} - h_{22})^2} \right]$$

Once you know λ, you find **x** by solving

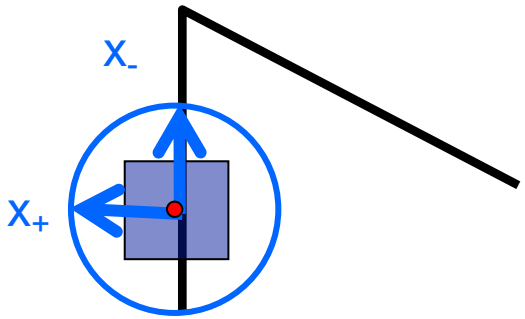
$$\begin{bmatrix} h_{11} - \lambda & h_{12} \\ h_{21} & h_{22} - \lambda \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 0$$

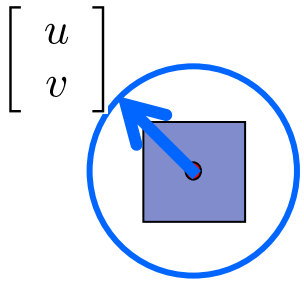
Source: S. Seitz.

Feature detection: the math

This can be rewritten:

$$E(u,v) = \sum_{(x,y)\in W} [u\ v] \underbrace{\begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}}_{H} \begin{bmatrix} u \\ v \end{bmatrix}$$





Eigenvalues and eigenvectors of H

- Define shifts with the smallest and largest change (E value)
- x_+ = direction of largest increase in E.
- λ_+ = amount of increase in direction x_+
- x_- = direction of smallest increase in E.
- λ_- = amount of increase in direction x_+

$$Hx_+ = \lambda_+ x_+$$
$$Hx_- = \lambda_- x_-$$

Source: S. Seitz.

Feature detection: the math

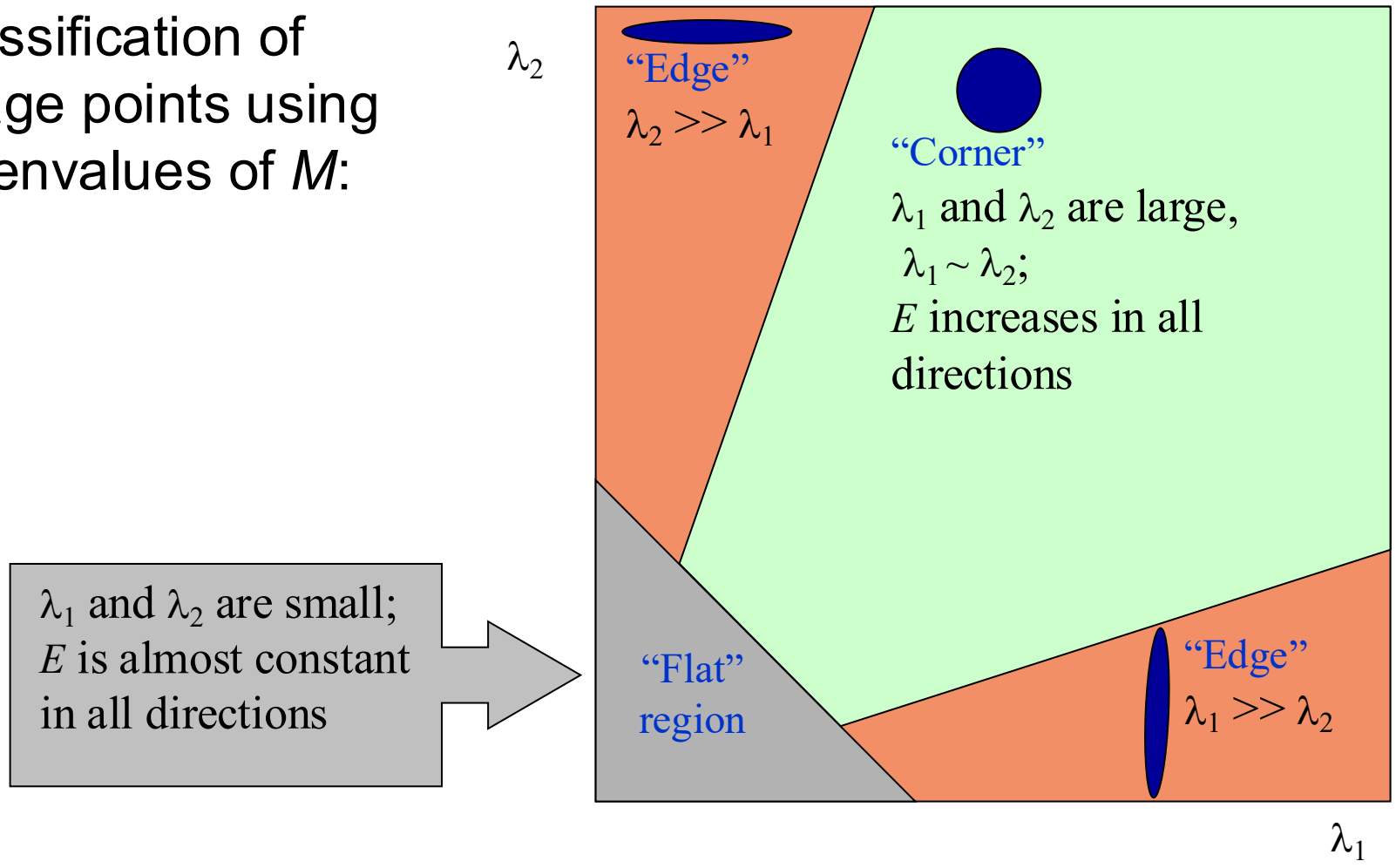
How are λ_+, x_+, λ_-, and x_+ relevant for feature detection?

- What's our feature scoring function?

Source: S. Seitz.

Feature detection: the math

Classification of image points using eigenvalues of *M*:



Source: Kokkinos, Saverese.

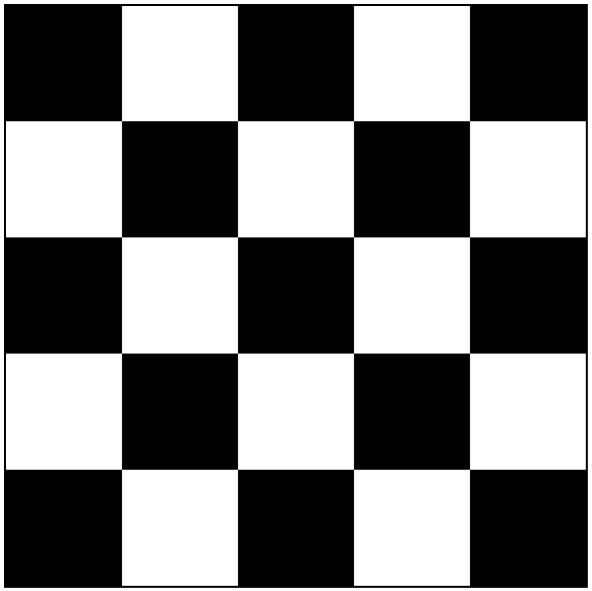
Feature detection: the math

How are λ_+, x_+, λ_-, 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 (λ_-) of *H*



I

What do these two images look like?

λ_+

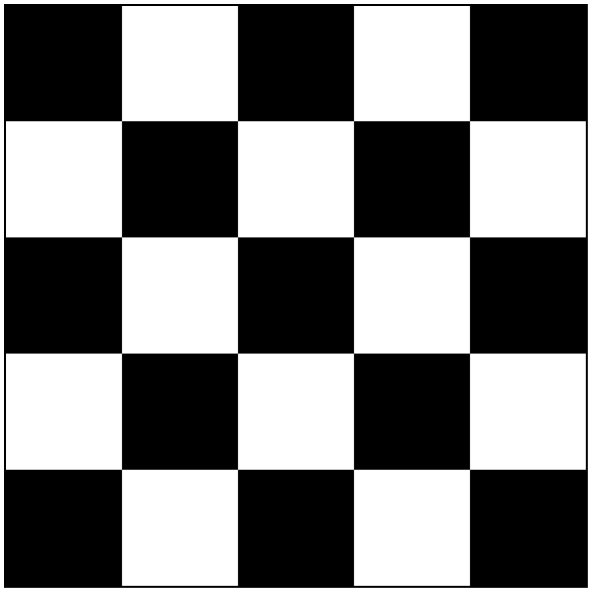
λ_-

Source: S. Seitz.

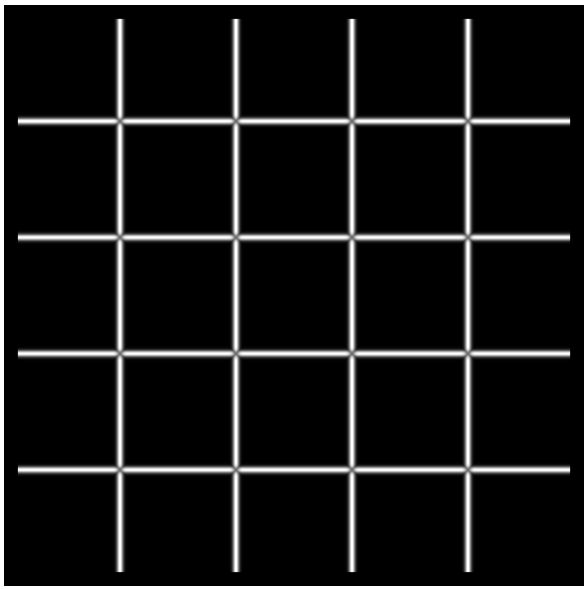
Feature detection summary

Here's what you do

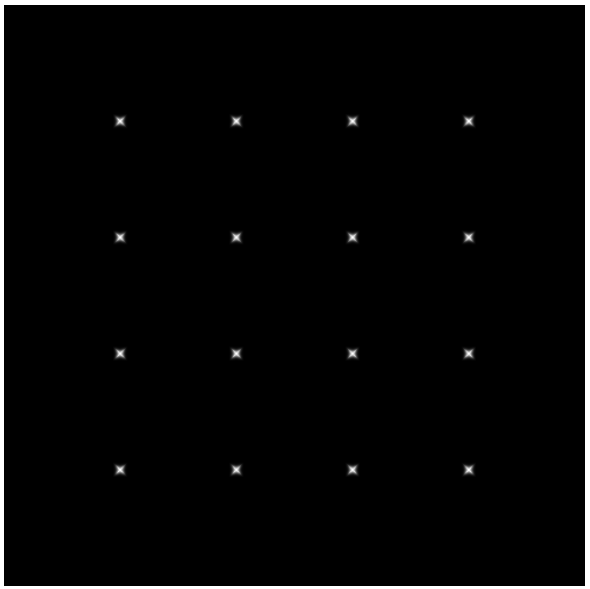
- Compute the gradient at each point in the image
- Create the *H* matrix from the entries in the gradient
- Compute the eigenvalues.
- Find points with large response (λ_- > threshold)
- Choose those points where λ_- is a local maximum as features



I



λ_+



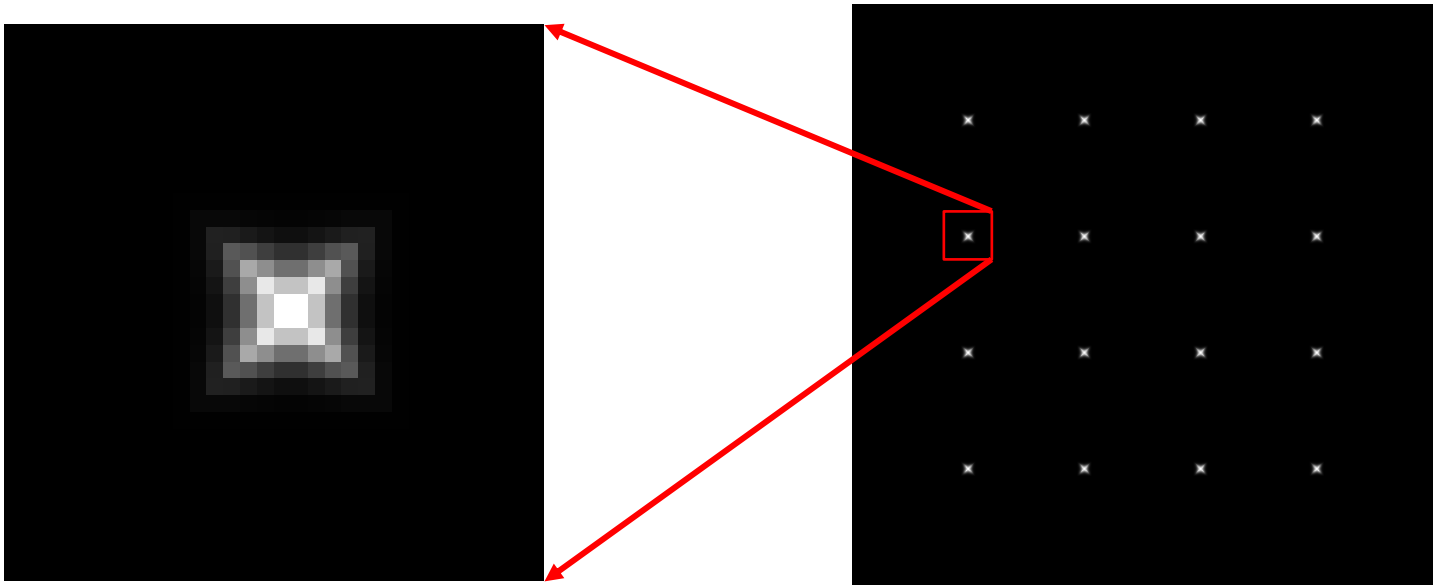
λ_-

Source: S. Seitz.

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 (λ_- > threshold)
- Choose those points where λ_- is a local maximum as features



λ_-

Source: S. Seitz.

The Harris operator

λ_- is a variant of the "Harris operator" for feature detection

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

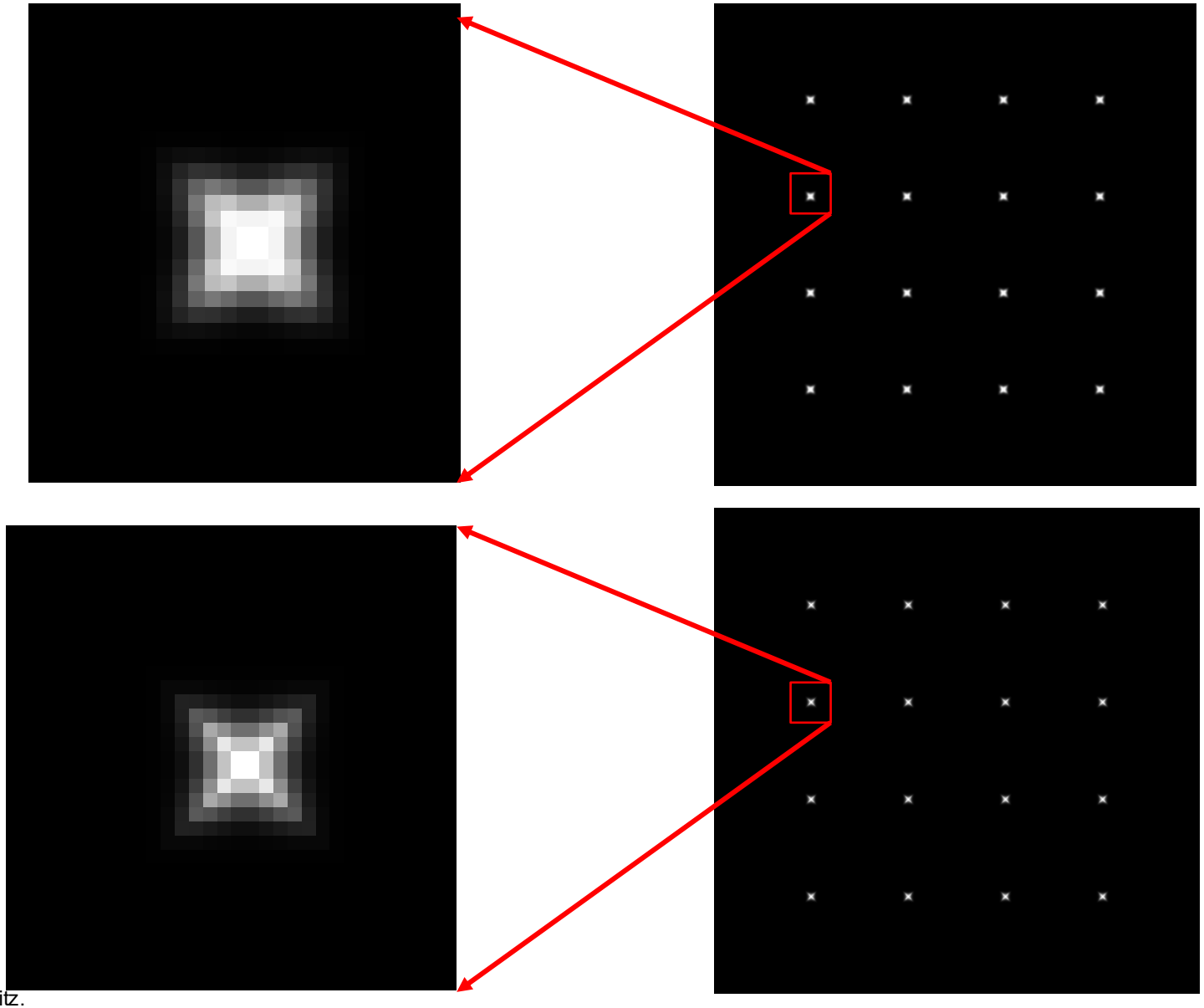
$$= \frac{determinant(H)}{trace(H)}$$

- The *trace* is the sum of the diagonals, i.e., *trace(H)* = $h_{11} + h_{22}$
- Very similar to λ_- 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.

Source: S. Seitz.

The Harris operator

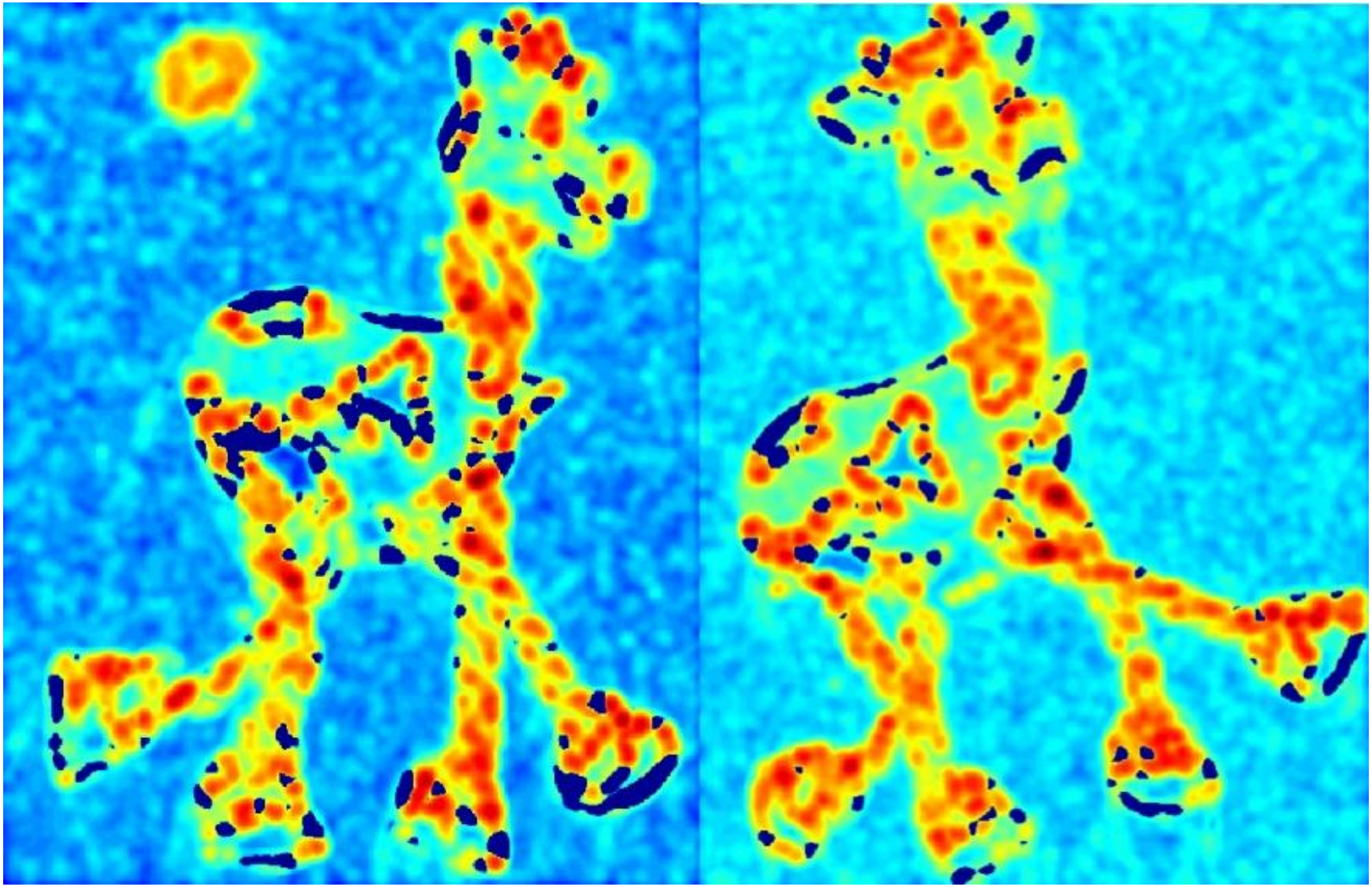


Source: S. Seitz.

Harris detector example

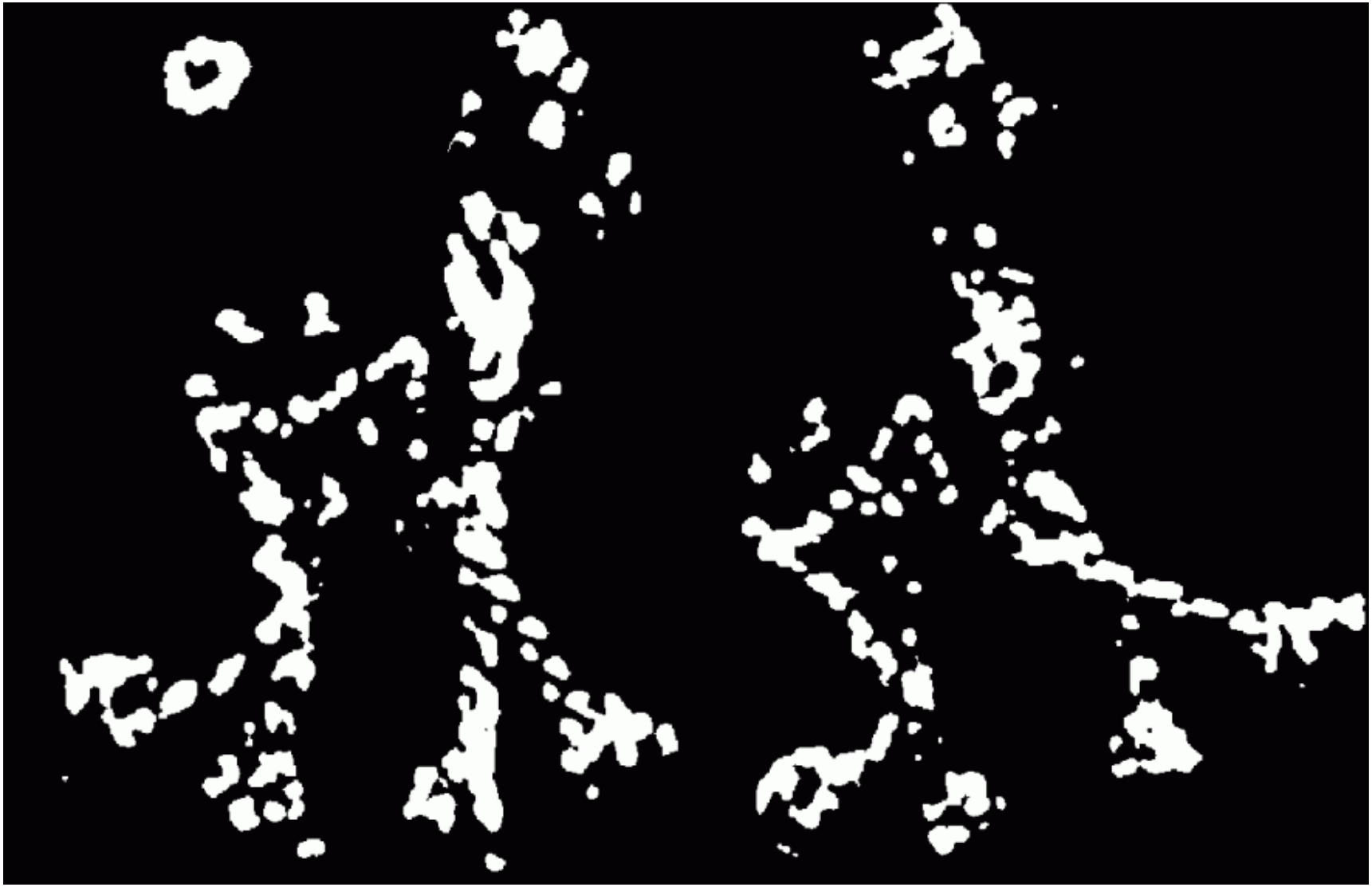
Source: S. Seitz.

f value (red high, blue low)



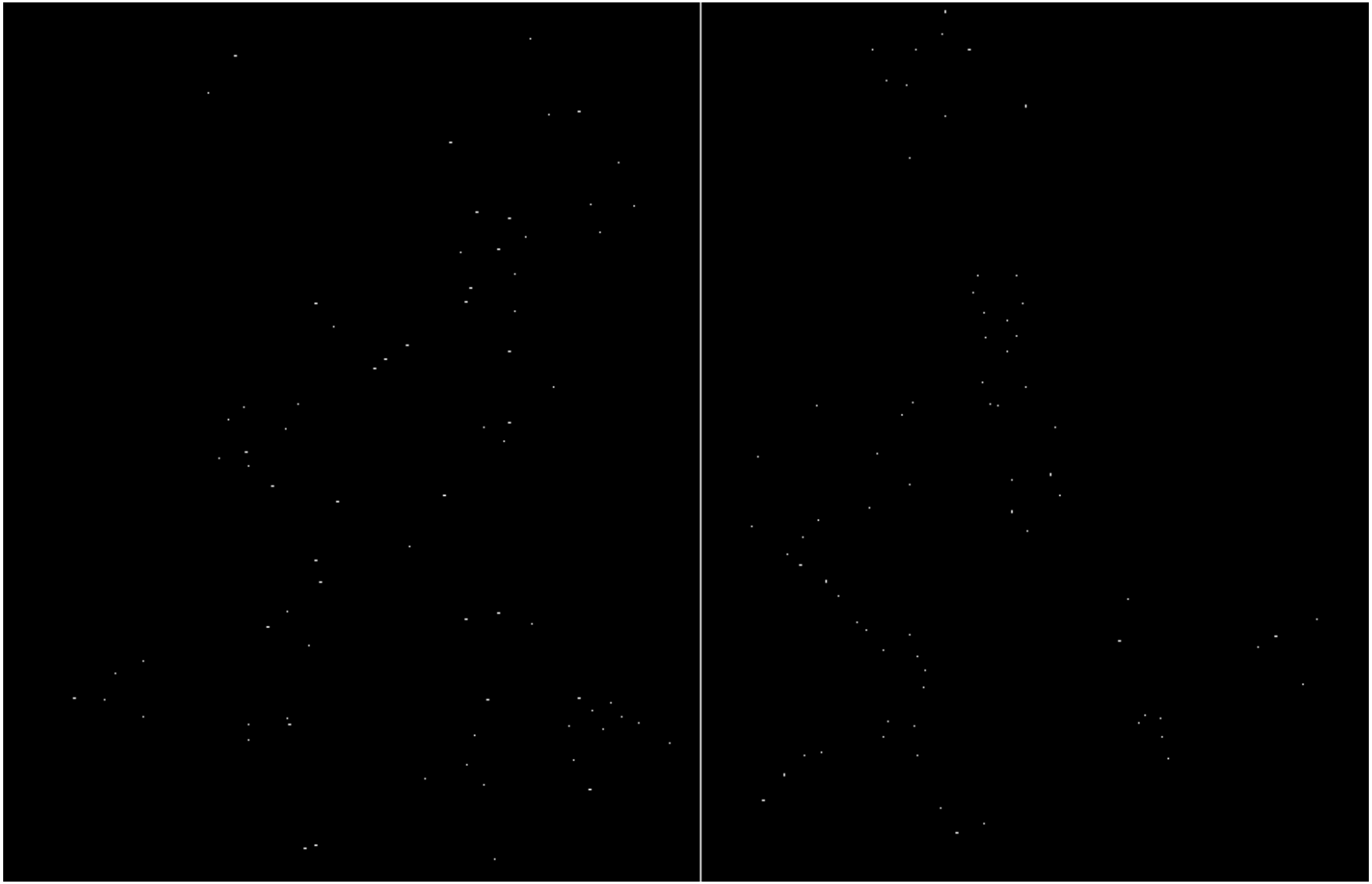
Source: S. Seitz.

Threshold (f > value)



Source: S. Seitz.

Find local maxima of f



Source: S. Seitz.

Harris features (in red)

Stop Slides

End of Corner Detector Lecture

(Invariance of Corner Detector Covered in PVL)

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.

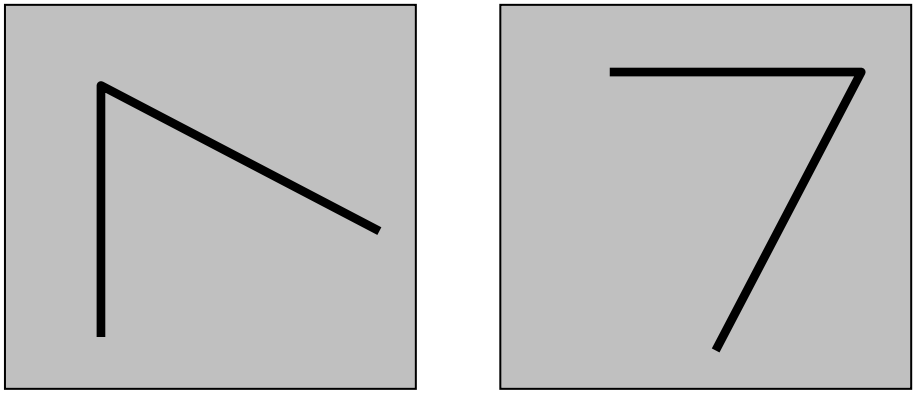
J may be a transformed version of I

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

Source: S. Seitz.

Harris Detector: Some Properties

- Is the Harris detector rotationally invariant?



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

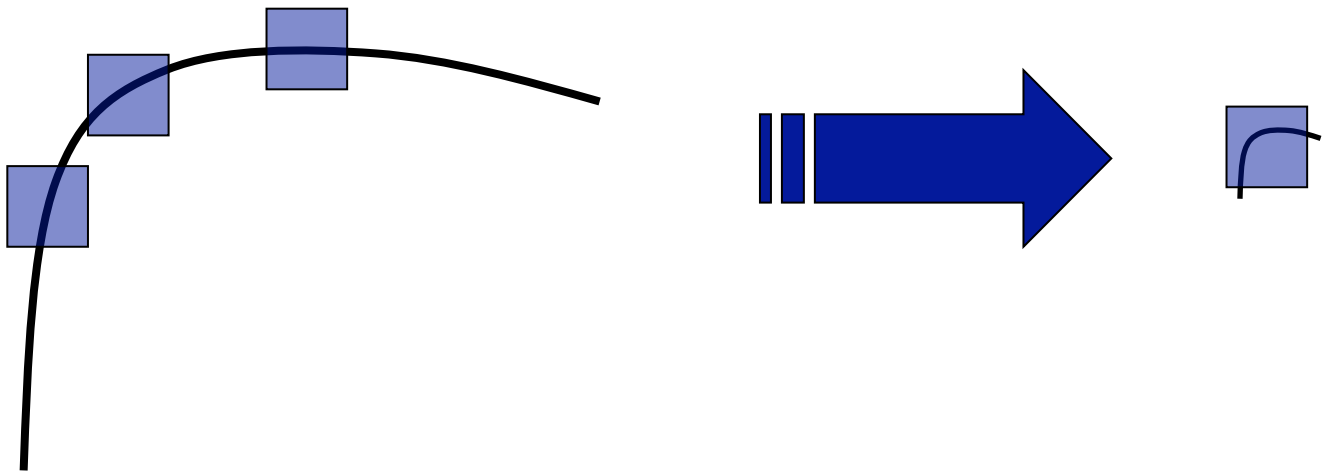
$$H = U^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} U \rightarrow f(\lambda_1, \lambda_2)$$ doesn't change!

Source: S. Savarese.

Harris Detector: Some Properties

- Is it scale invariant?

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



All points will be classified as edges

Corner !

Source: S. Savarese.

Harris Detector: Some Properties

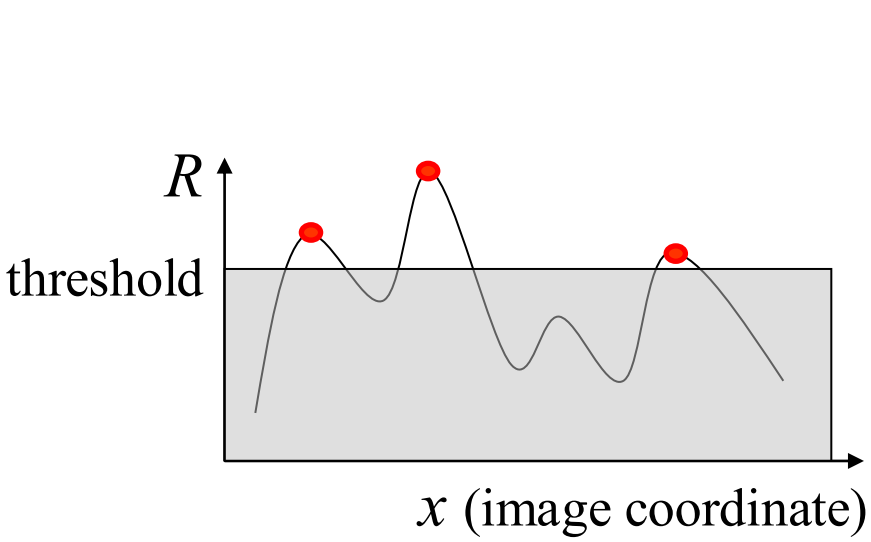
- Partial invariance to *affine intensity* changes

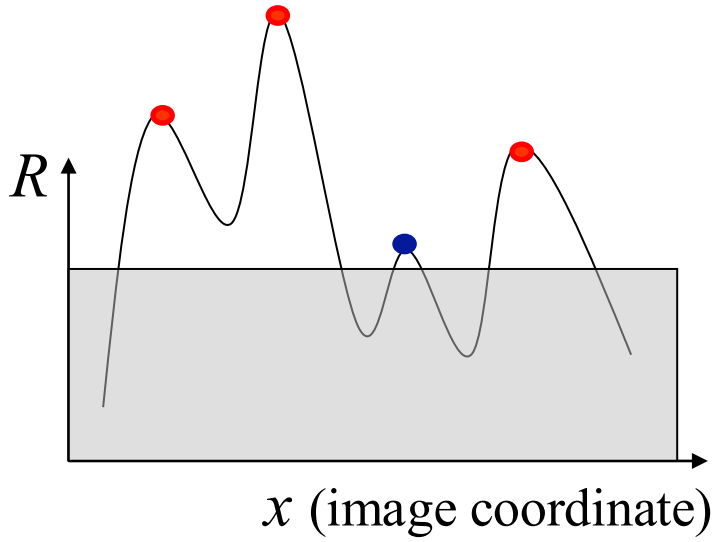
$$I \rightarrow s\, I + b$$

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

    (only derivatives are used)

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





Source: S. Savarese.

Invariance

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

Source: S. Savarese.

Next Steps (After Initial Discussion on Description)

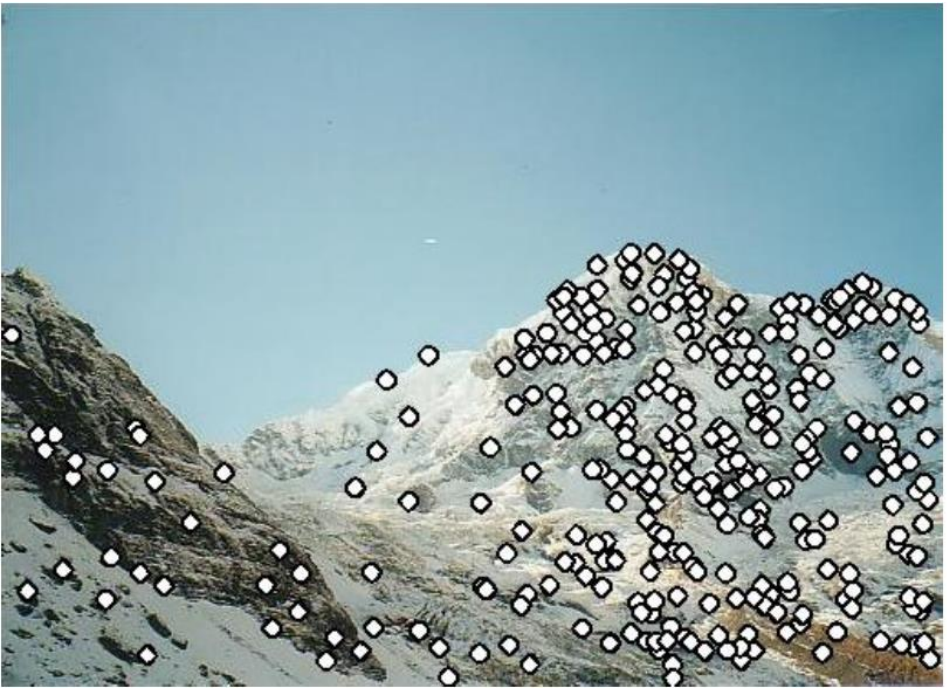
- Exploring further invariance in feature detection
  - Scale invariance, scale-space and adaptive scale selection
  - Affine invariance
- Exploring further invariance in feature description
  - Photometric invariance
  - Rotation invariance (noted in this lecture)
  - Affine invariance

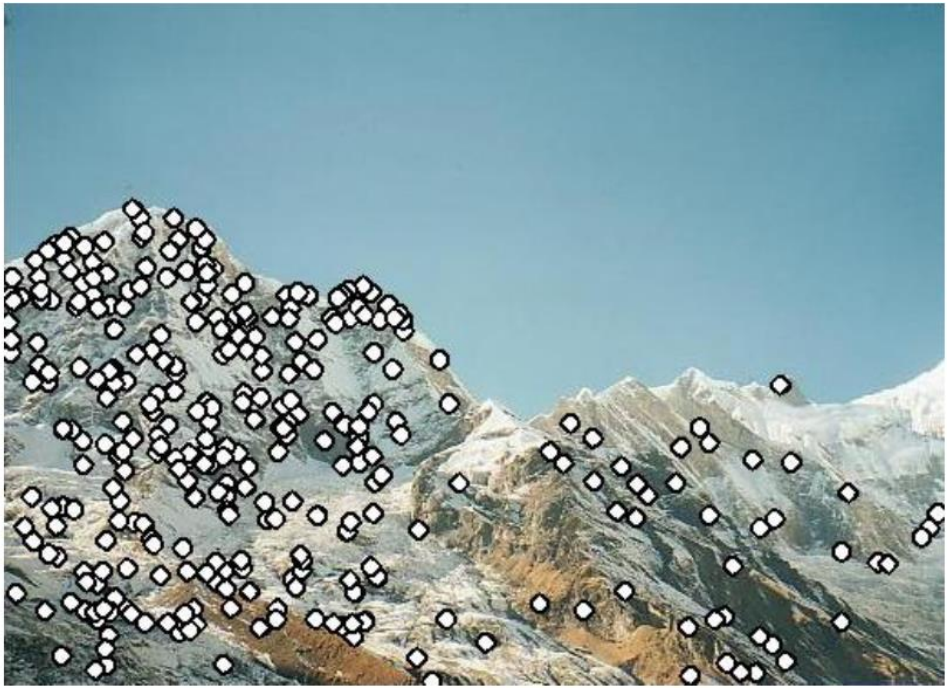
Local Feature Descriptors

Application: Image Stitching

Sources for this example: I. Kokkinos, D. Frolova, D. Simakov.

Application: Image Stitching

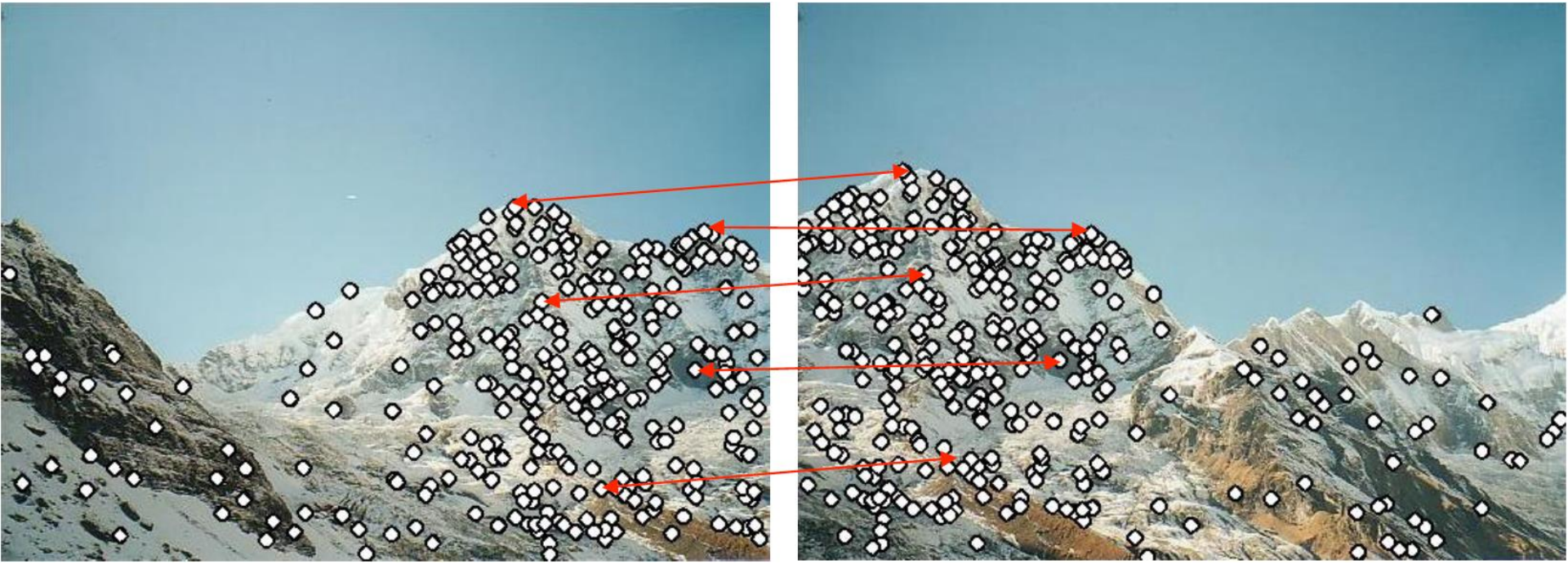




1. Detect feature points in both images.

Sources for this example: I. Kokkinos, D. Frolova, D. Simakov.

Application: Image Stitching



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

Sources for this example: I. Kokkinos, D. Frolova, D. Simakov.

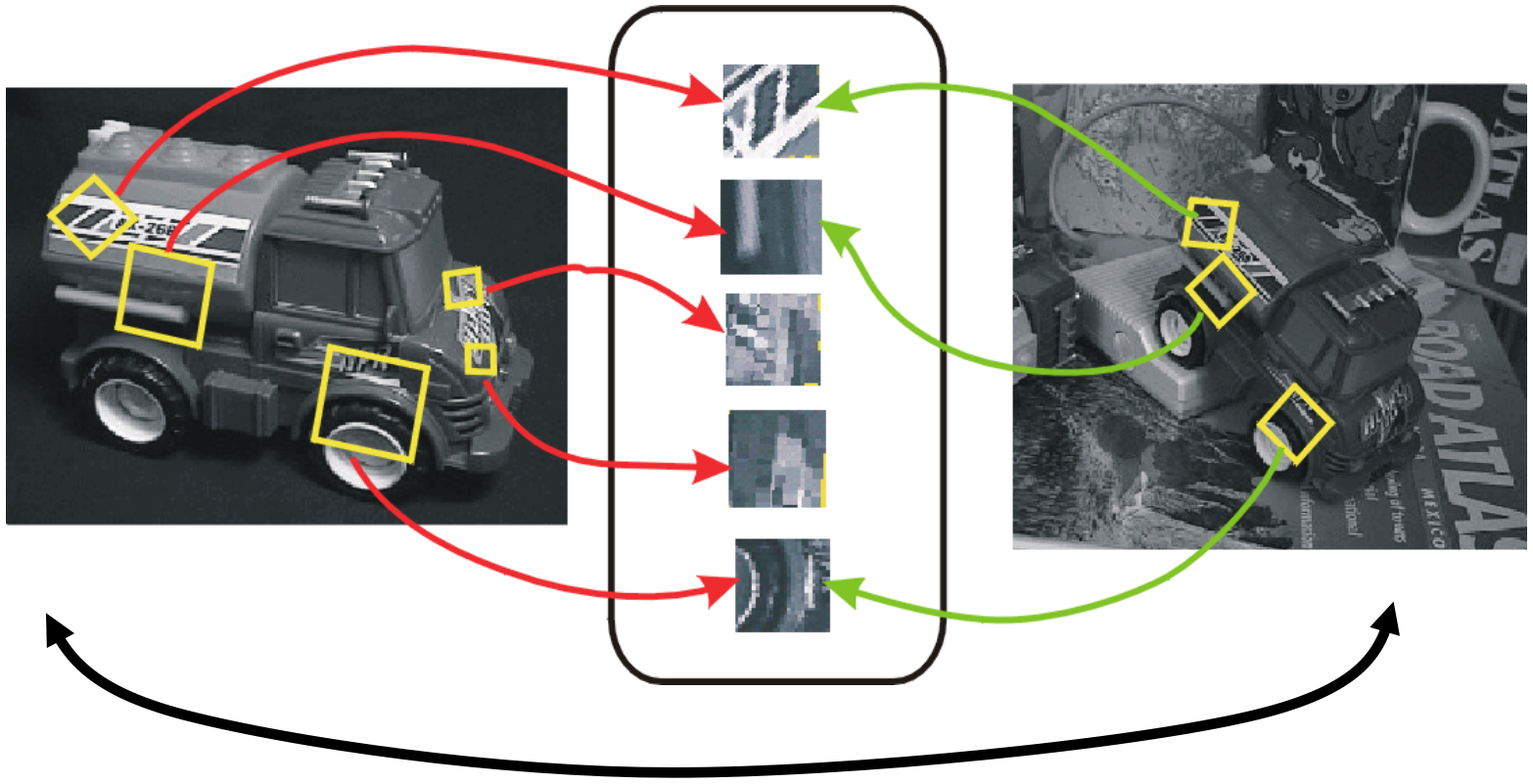
Application: Image Stitching

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

Sources for this example: I. Kokkinos, D. Frolova, D. Simakov.

Pose normalization

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

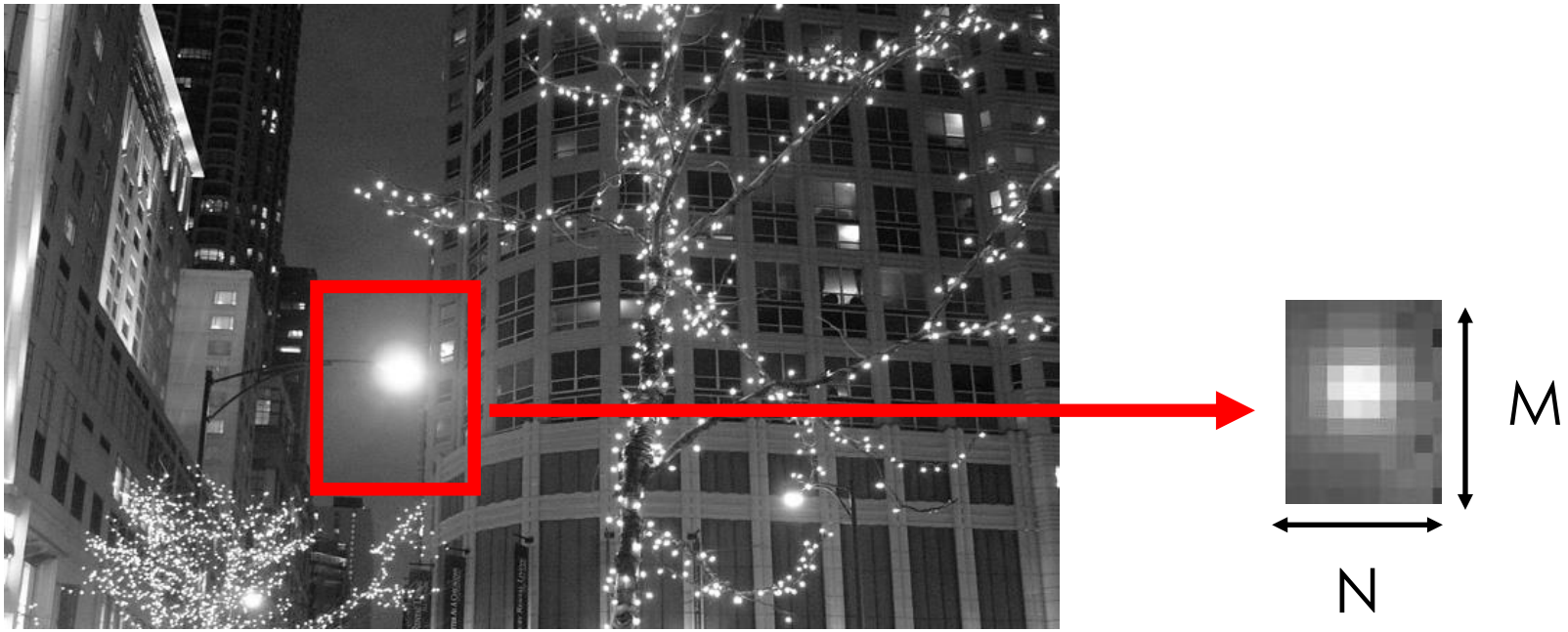


Courtesy of D. Lowe

Change of scale, pose, illumination…

Source: S. Savarese.

The simplest descriptor



1 x NM vector of pixel intensities

w= […]

$$\mathbf{w}_n = \frac{(\mathbf{w} - \overline{\mathbf{w}})}{\left\|(\mathbf{w} - \overline{\mathbf{w}})\right\|}$$

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

Source: S. Savarese.

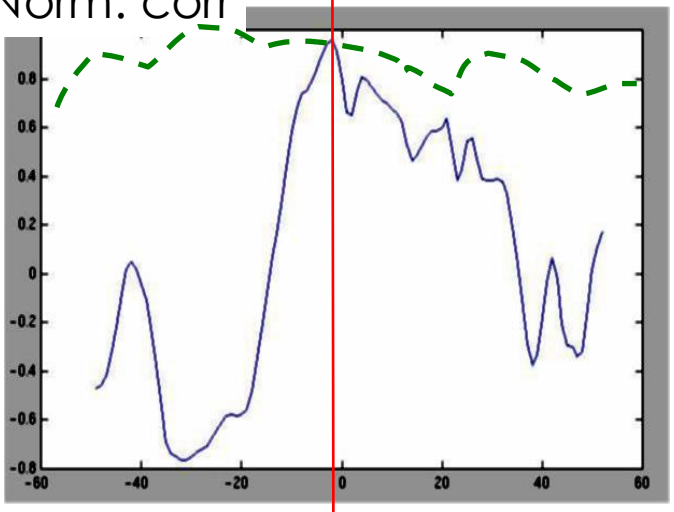
Why not?

- Sensitive to small variation of:
  - Location
  - Pose
  - Scale
  - Intra-class variability
- Poorly distinctive

Source: S. Savarese.

Sensitive to pose variations

Norm. corr



Normalized Correlation:

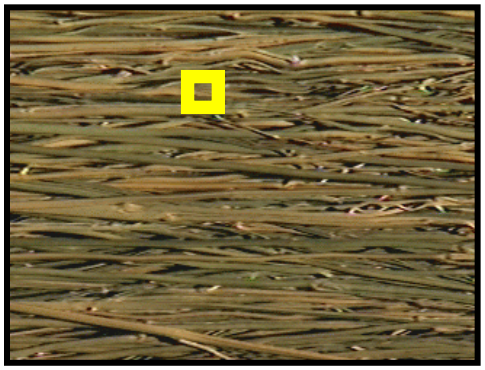
$$w_n \cdot w'_n = \frac{(w - \overline{w})(w' - \overline{w}')}{\left\|(w - \overline{w})(w' - \overline{w}')\right\|}$$

Source: S. Savarese.

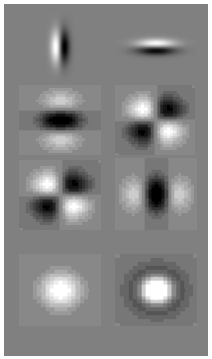
Detector	Illumination	Pose	Intra-class variab.
PATCH	Good	Poor	Poor

Source: S. Savarese.

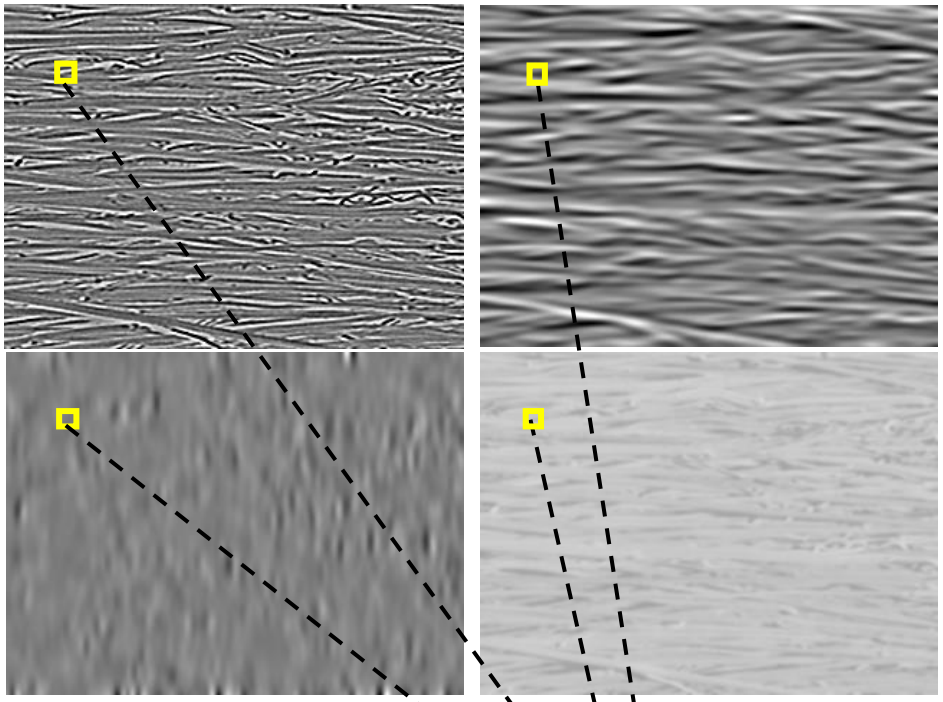
Bank of filters



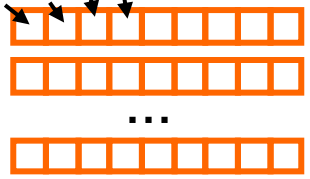
image



filter bank



filter responses



descriptor

More robust but still quite sensitive to pose variations

Source: S. Savarese.

Detector	Illumination	Pose	Intra-class variab.
PATCH	Good	Poor	Poor
FILTERS	Good	Medium	Medium

Source: S. Savarese.

SIFT descriptor

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

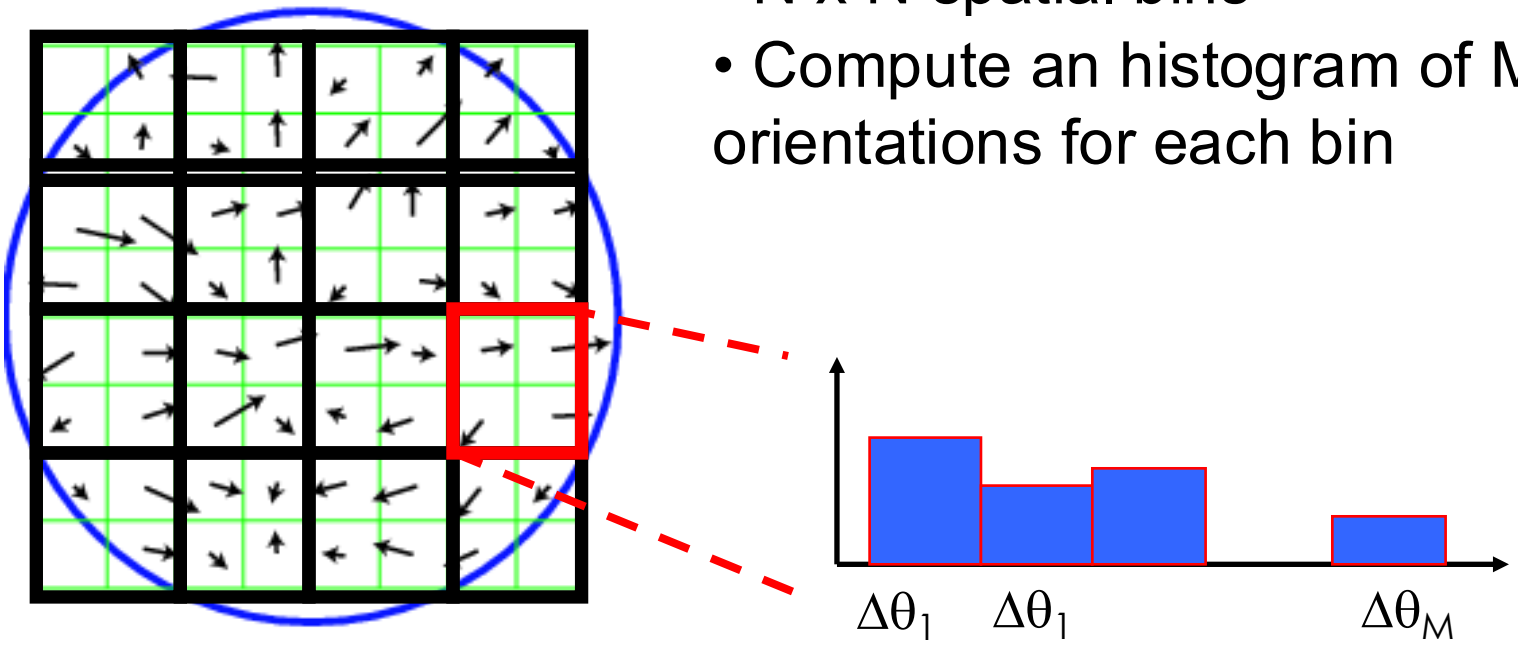
- Alternative representation for image patches
- Location and characteristic scale s given by DoG detector

Source: S. Savarese.

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



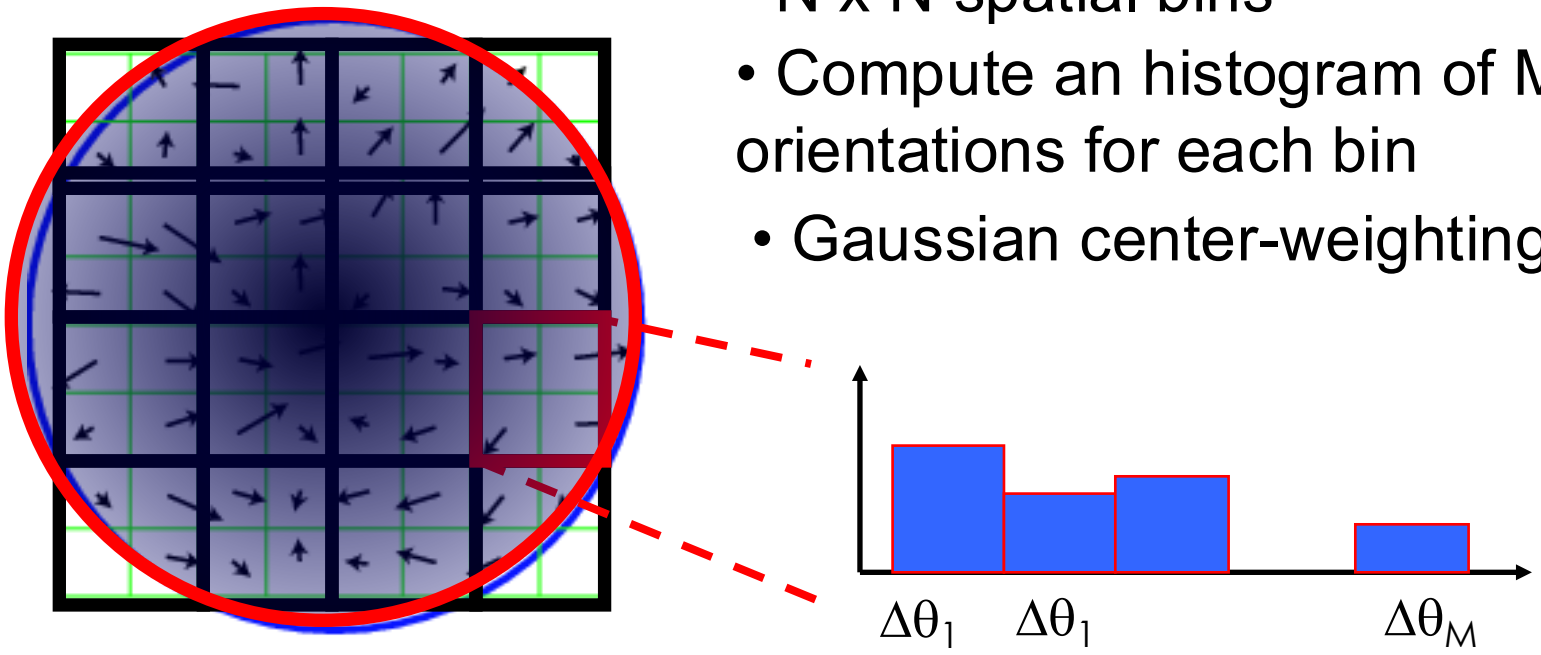
- Compute gradient at each pixel
- N x N spatial bins
- Compute an histogram of M orientations for each bin

Source: S. Savarese.

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



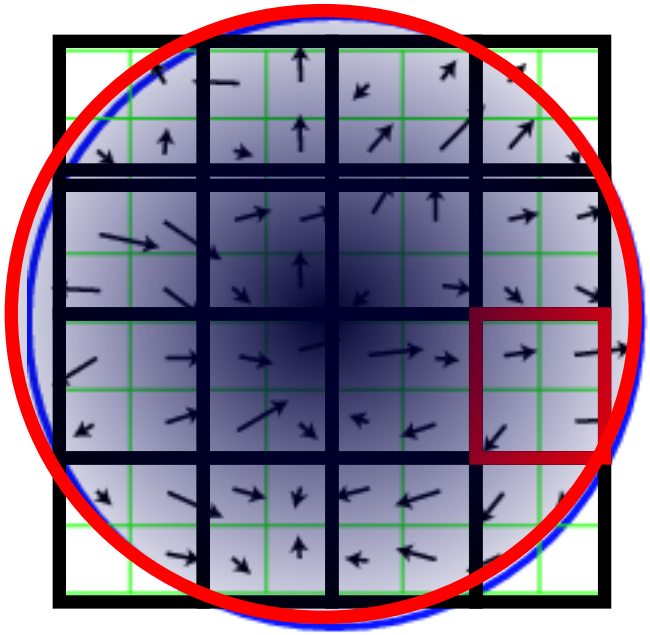
- Compute gradient at each pixel
- N x N spatial bins
- Compute an histogram of M orientations for each bin
- Gaussian center-weighting

Source: S. Savarese.

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 x N spatial bins
- Compute an histogram of M orientations for each bin
- Gaussian center-weighting
- Normalized unit norm

Typically M = 8; N= 4
1 x 128 descriptor

Source: S. Savarese.

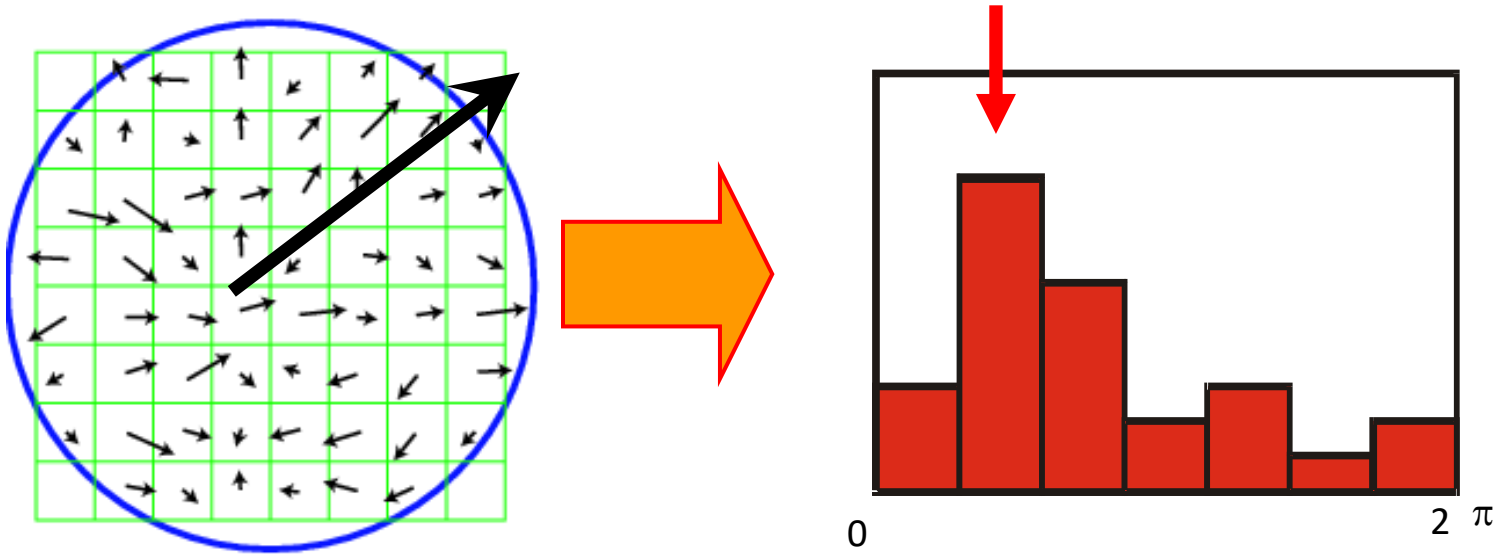
SIFT Descriptor

- 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)

Source: S. Savarese.

Rotational Invariance

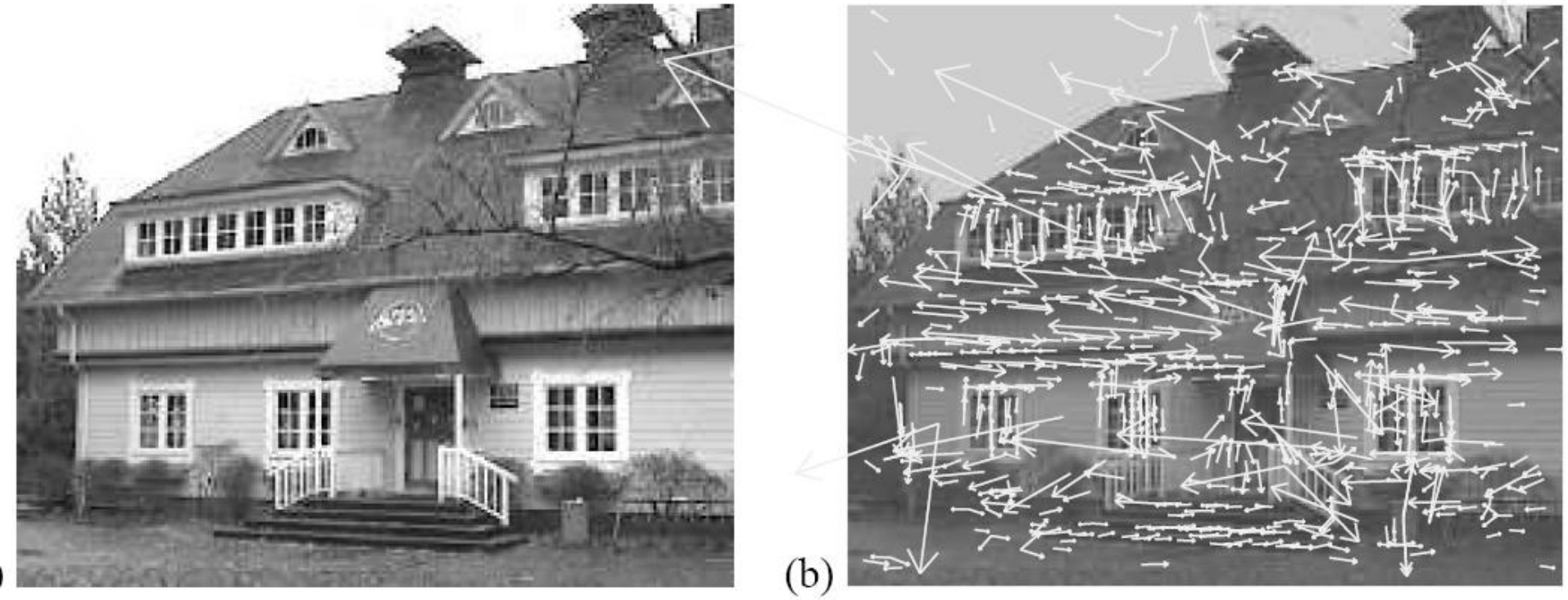
- Find dominant orientation by building smoothed orientation histogram
- Rotate all orientations by the dominant orientation



This makes the SIFT descriptor rotational invariant

Source: S. Savarese.

SIFT Rotational Invariance Example



Source: S. Savarese.

Rotation invariance (Alternate)

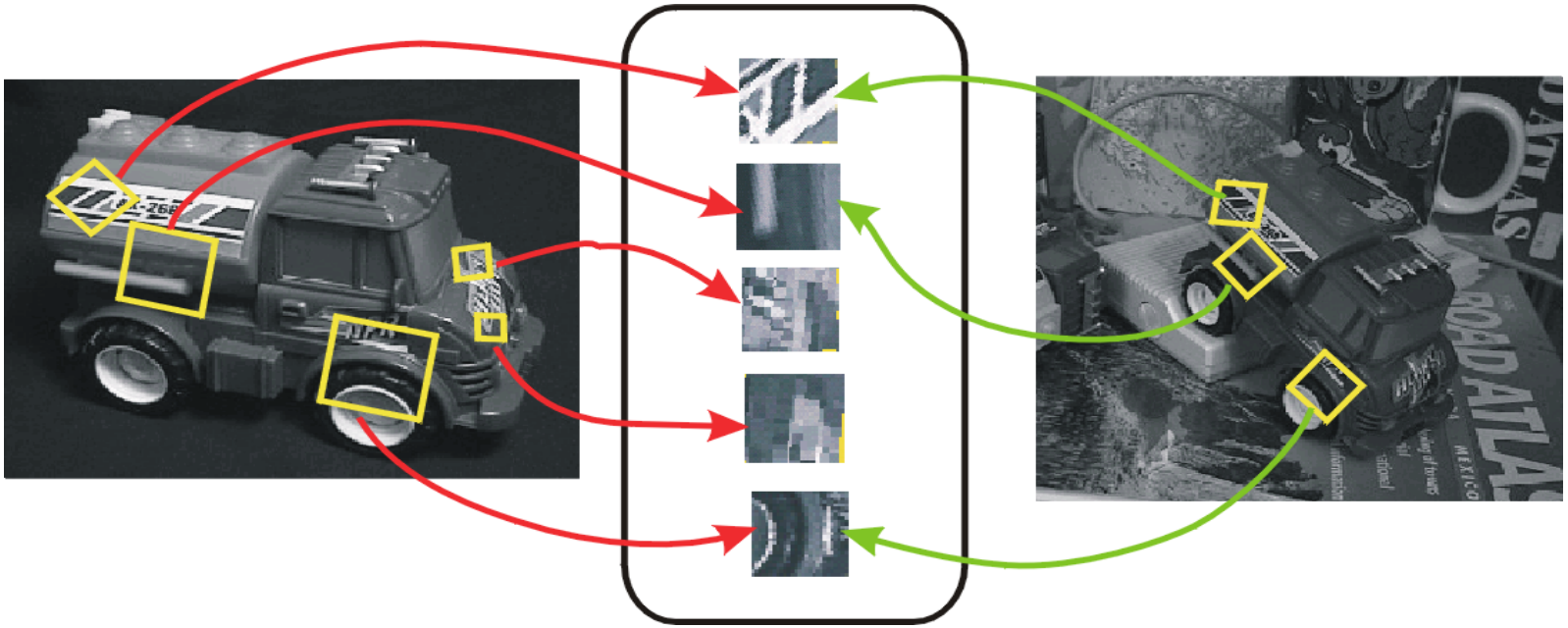
Find dominant orientation of the image patch

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

Figure by Matthew Brown

Source: S. Seitz.

SIFT Rotational Invariance Example



Source: S. Savarese.

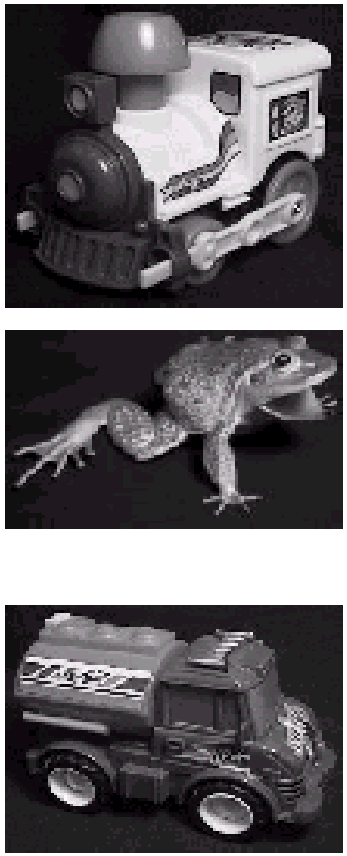
Matching Using SIFT

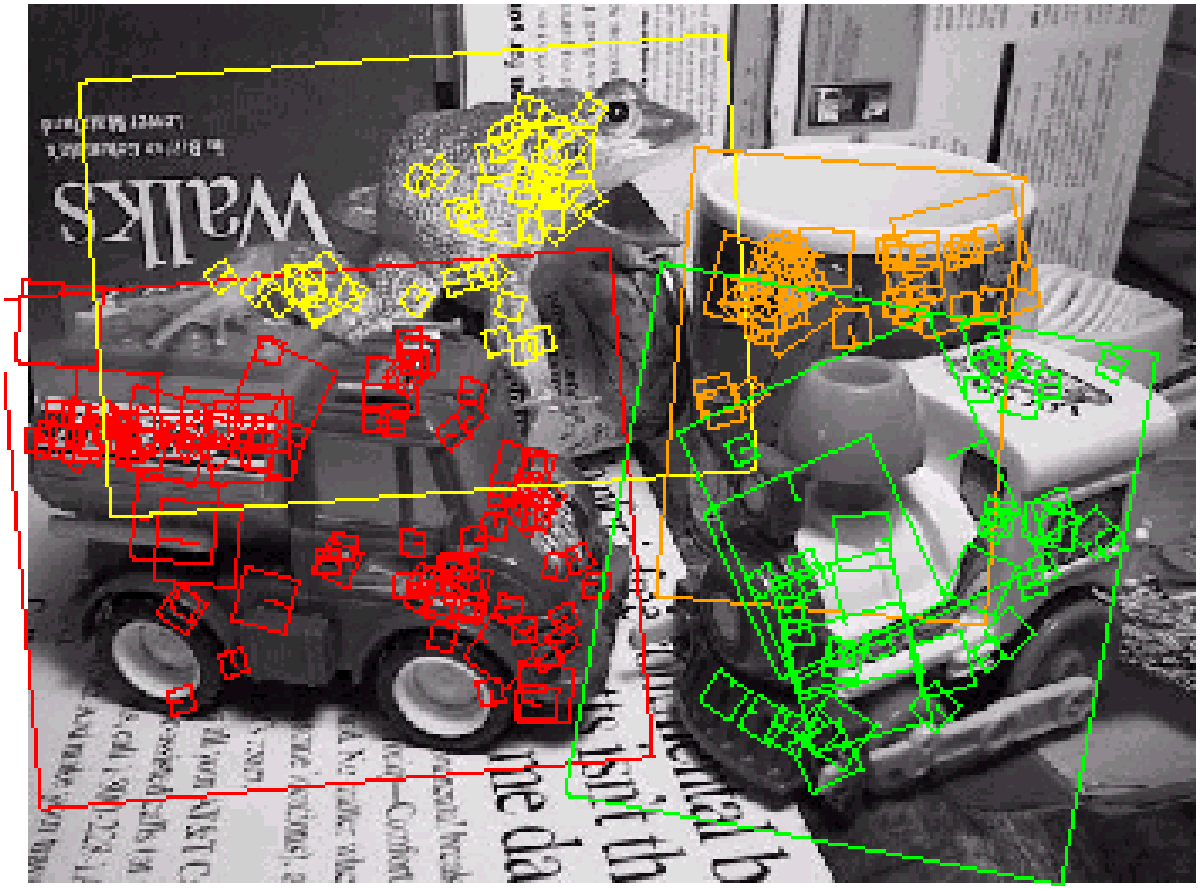
David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), 04

Source: S. Savarese.

Matching Using SIFT

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





Source: S. Savarese.

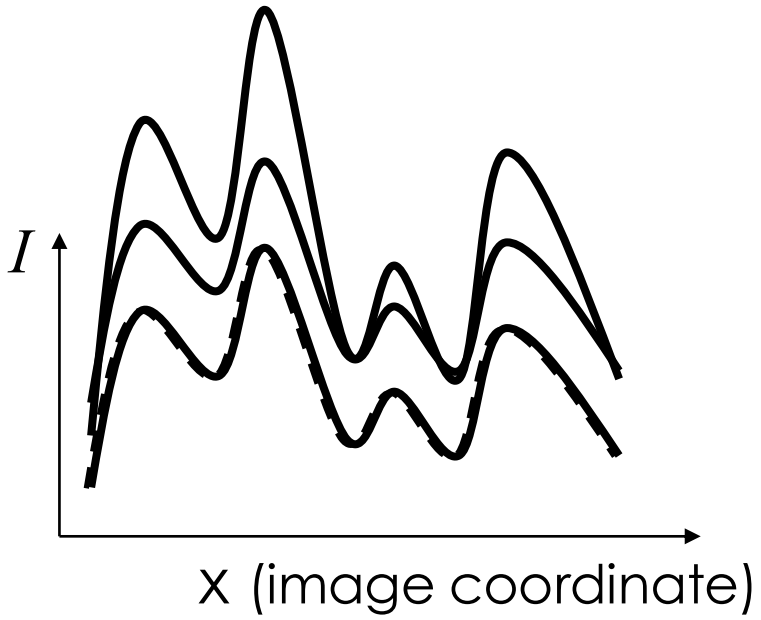
Detector	Illumination	Pose	Intra-class variab.
PATCH	Good	Poor	Poor
FILTERS	Good	Medium	Medium
SIFT	Good	Good	Medium

Source: S. Savarese.

Illumination normalization

- *Affine intensity* change:

$$I \rightarrow I + b$$
$$\rightarrow a\,I + b$$



•Make each patch zero mean:

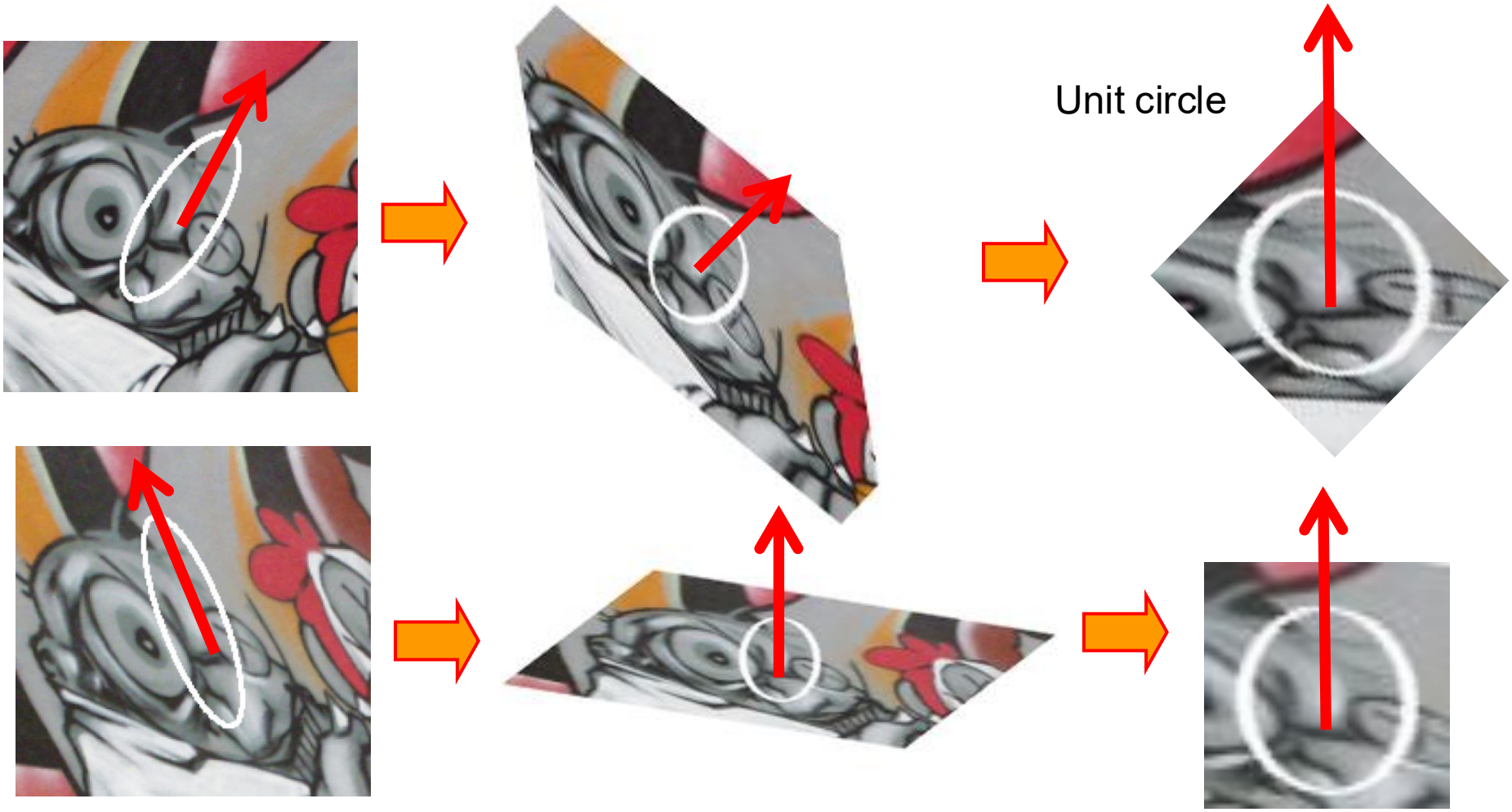
$$\mu = \frac{1}{N}\sum_{x,y} I(x,y)$$
$$Z(x,y) = I(x,y) - \mu$$

•Then make unit variance:

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

Source: S. Savarese.

Pose normalization



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

Source: S. Savarese.

Video Detectors / Features

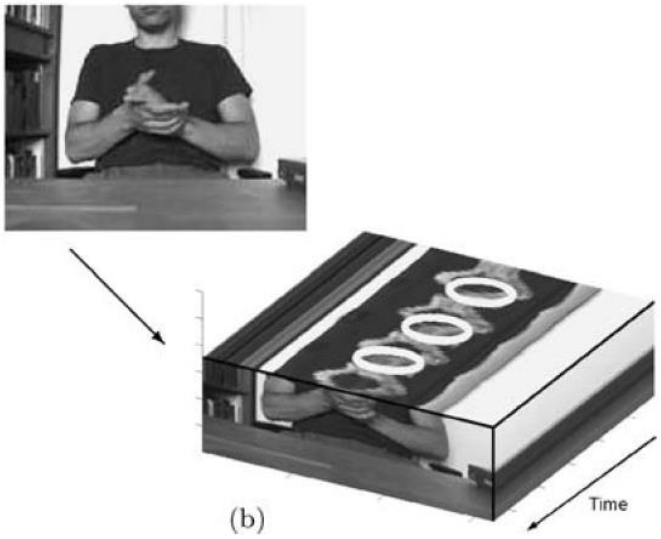
STIP: Space-Time Interest Points

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.

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

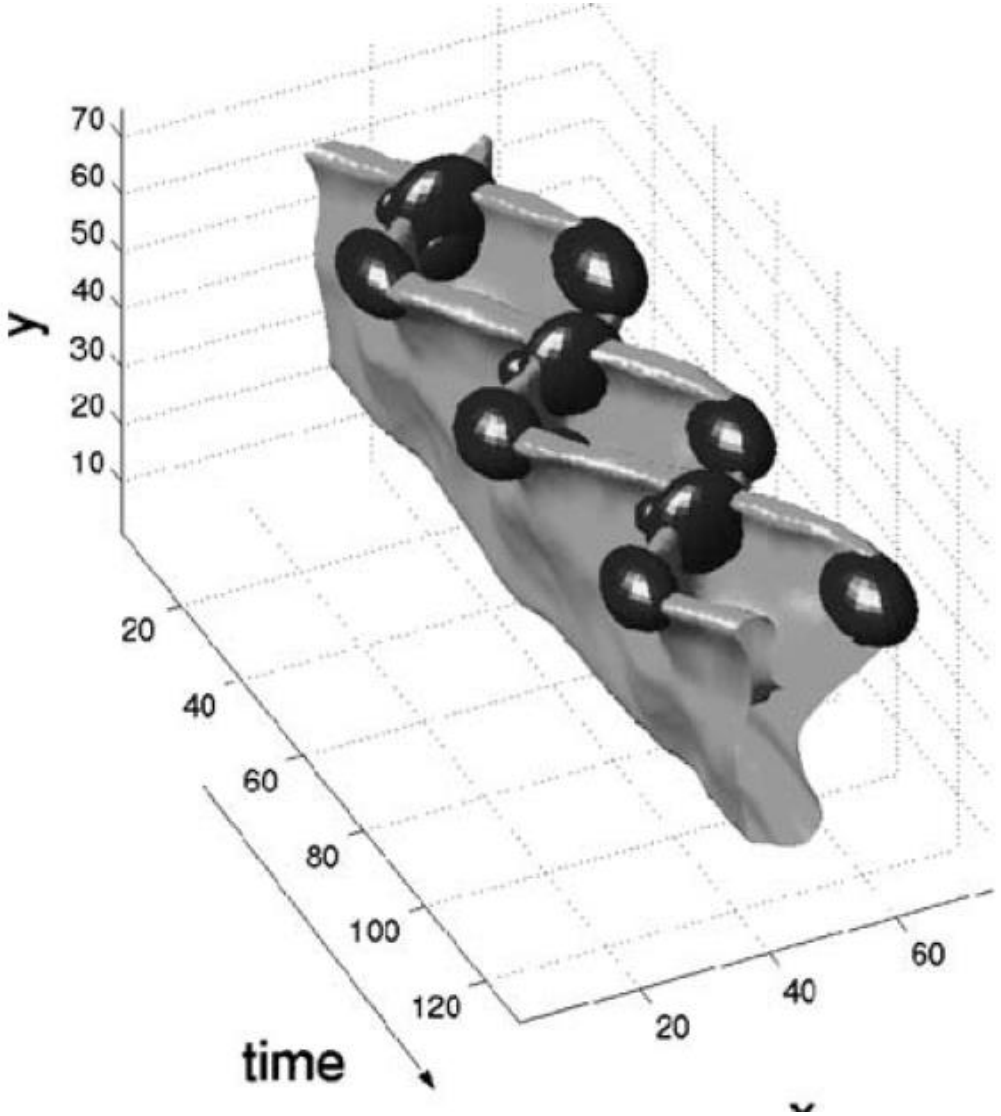
$$H = \det(\mu) - k \operatorname{trace}^2(\mu)$$



- 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.



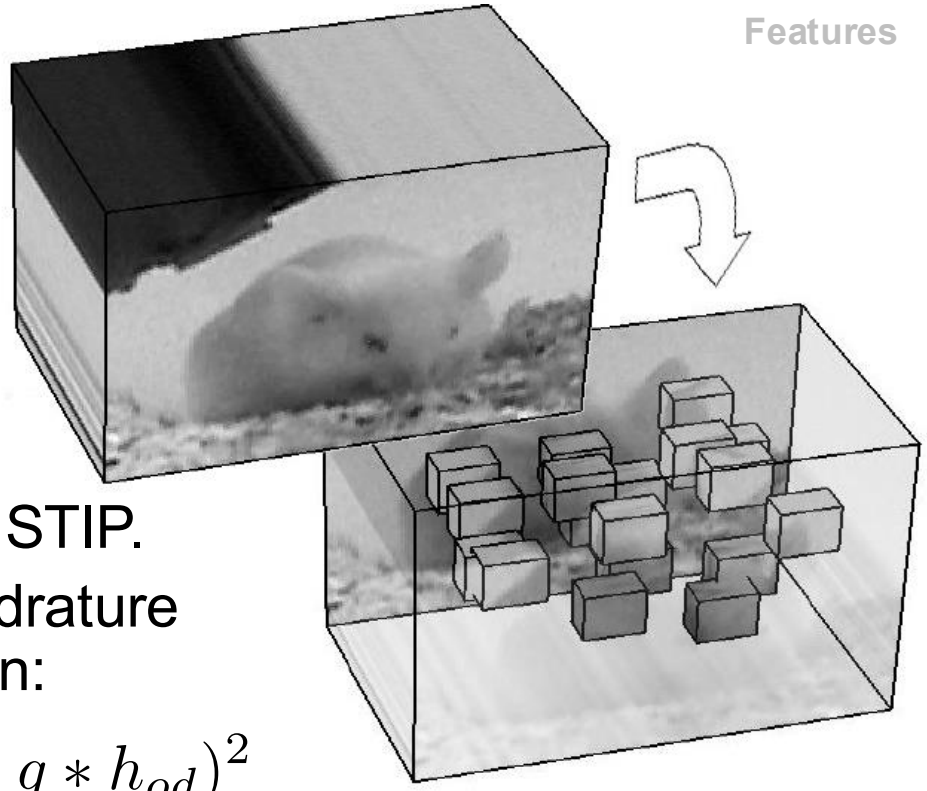
STIP: Space-Time Interest Points

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.

Dollár's Cuboids

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

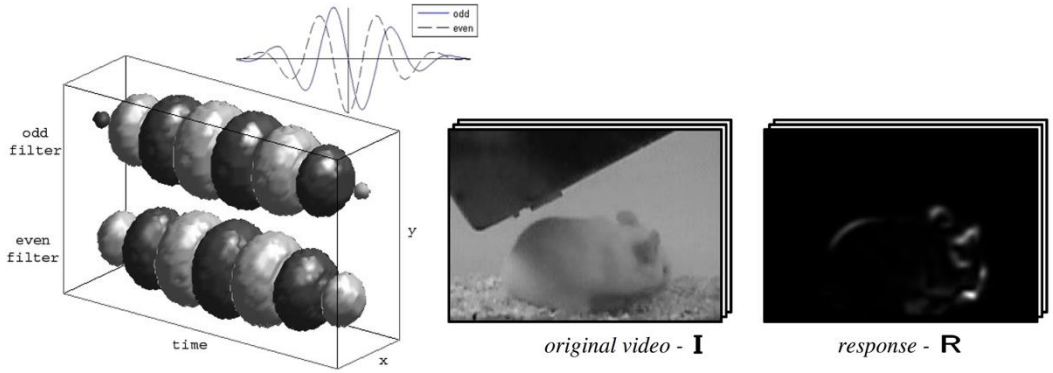


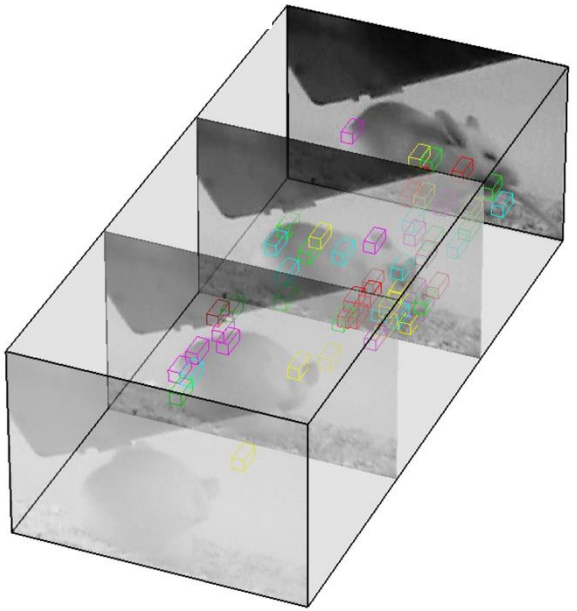
- Detector fires when local image intensities contain periodic frequency components.
- It will fire more frequently than STIP.
- Based on temporal Gabor quadrature pair filter with response function:

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

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

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





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*.

Discussion: Local Spatiotemporal Features

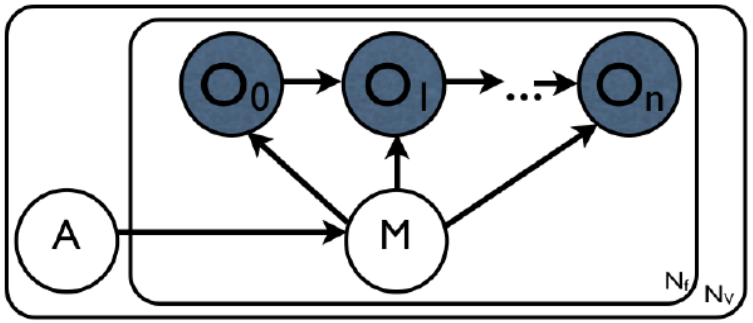
- Benefits of local feature methods:
  - Robustness to viewpoint changes and occlusion.
  - Relatively computationally inexpensive.
  - Do not need to detect and track the agent.
  - Implicitly incorporate motion, form, and context.

- 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:

$$P(A,O) = \sum_M P(A,M,O) =$$

$$P(A)\prod_f^{N_f}\sum_i^{N_m} P(M_f^i|A)P(O_{0,f}|M_f^i)$$

$$\prod_{t=1}^{T_f} P(O_{t,f}|O_{t-1,f},M_f^i)$$

- 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

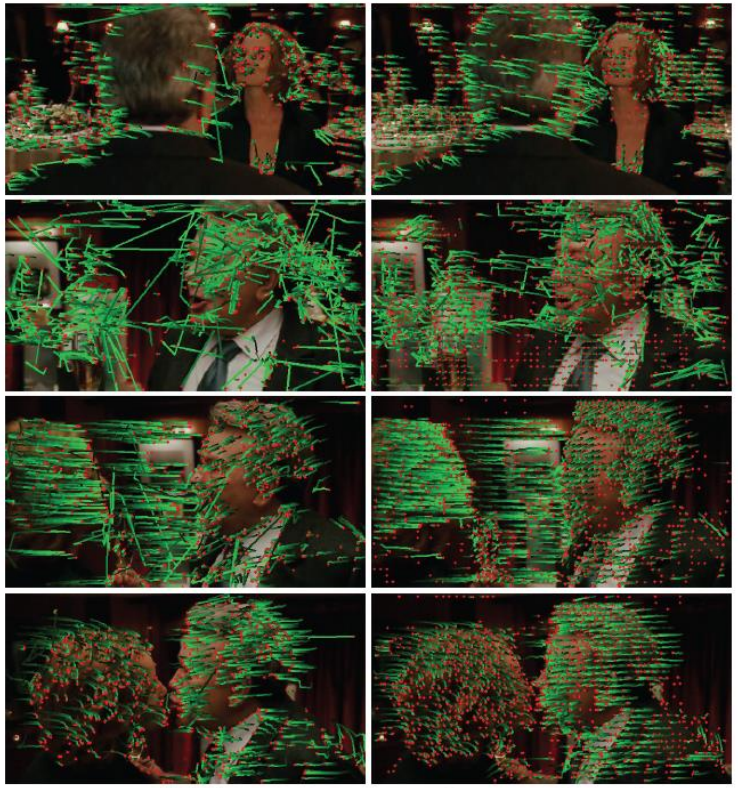
Trajectories by Local Keypoint Tracking

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

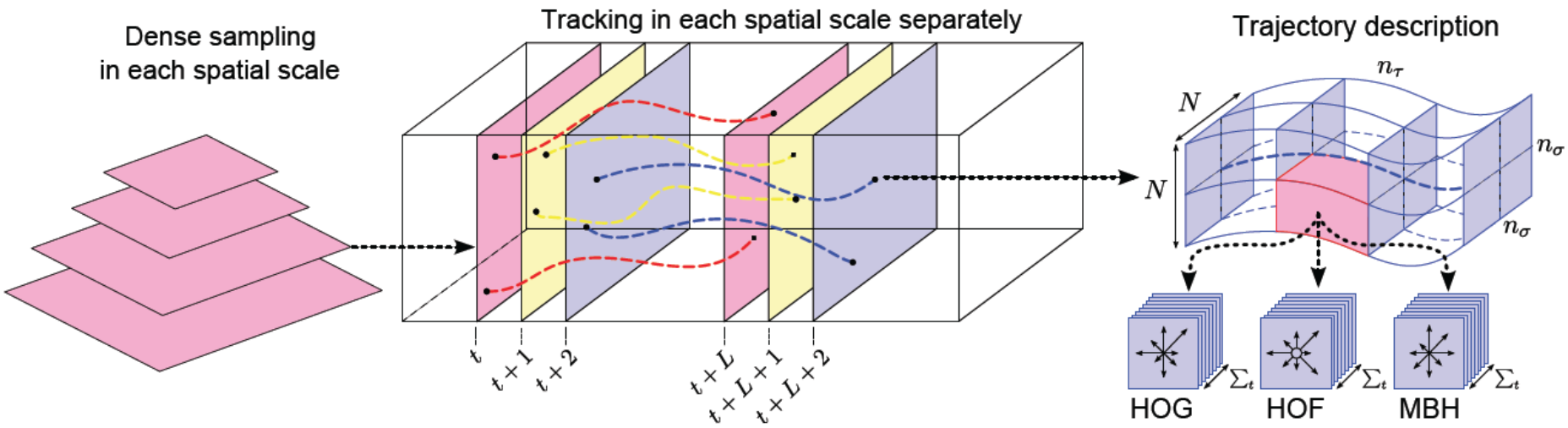
Dense Trajectories

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	KLT	Dense trajectories	KLT	Dense trajectories	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	**93.4%**	**94.2%**	**79.9%**	**84.2%**	**54.6%**	**58.3%**	82.1%	**88.2%**

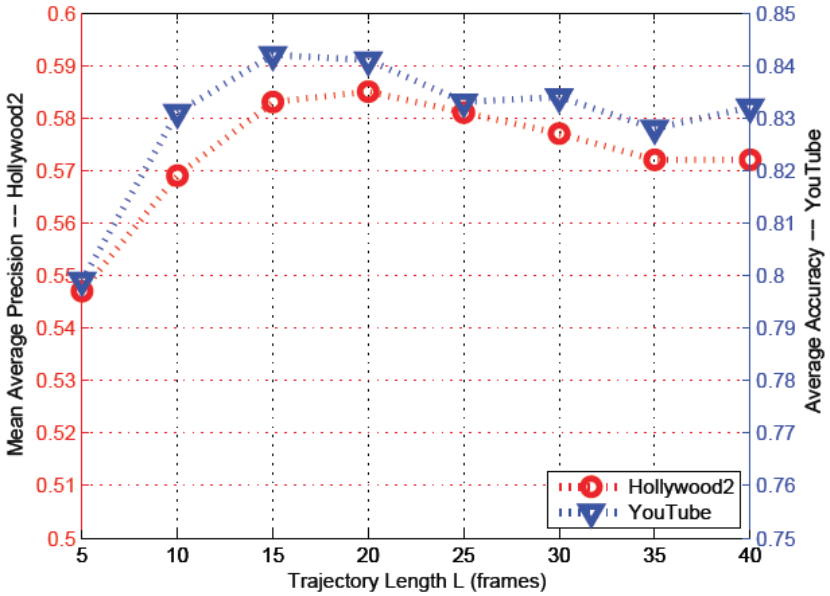
Table 1. Comparison of KLT and dense trajectories as well as different descriptors on KTH, YouTube, Hollywood2 and UCF sports. We report average accuracy over all classes for KTH, YouTube and UCF sports and mean AP over all classes for Hollywood2.

KTH		YouTube		Hollywood2		UCF sports	
Laptev *et al.* [14]	91.8%	Liu *et al.* [16]	71.2%	Wang *et al.* [32]	47.7%	Wang *et al.* [32]	85.6%
Yuan *et al.* [35]	93.3%	Ikizler-Cinbis *et al.* [9]	75.21%	Gilbert *et al.* [8]	50.9%	Kovashka *et al.* [12]	87.27%
Gilbert *et al.* [8]	94.5%			Ullah *et al.* [31]	53.2%	Kläser *et al.* [10]	86.7%
Kovashka *et al.* [12]	**94.53%**			Taylor *et al.* [29]	46.6%		
Our method	94.2%	Our method	**84.2%**	Our method	**58.3%**	Our method	**88.2%**

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%	**48.48%**
bike	87.6%	**91.7%**	75.17%
dive	**99.0%**	**99.0%**	95.0%
golf	95.0%	**97.0%**	95.0%
h_ride	76.0%	**85.0%**	73.0%
s_juggle	65.0%	**76.0%**	53.0%
swing	86.0%	**88.0%**	66.0%
t_swing	71.0%	71.0%	**77.0%**
t_jump	93.0%	**94.0%**	93.0%
v_spike	**96.0%**	95.0%	85.0%
walk	76.4%	**87.0%**	66.67%
Accuracy	79.9%	**84.2%**	75.21%

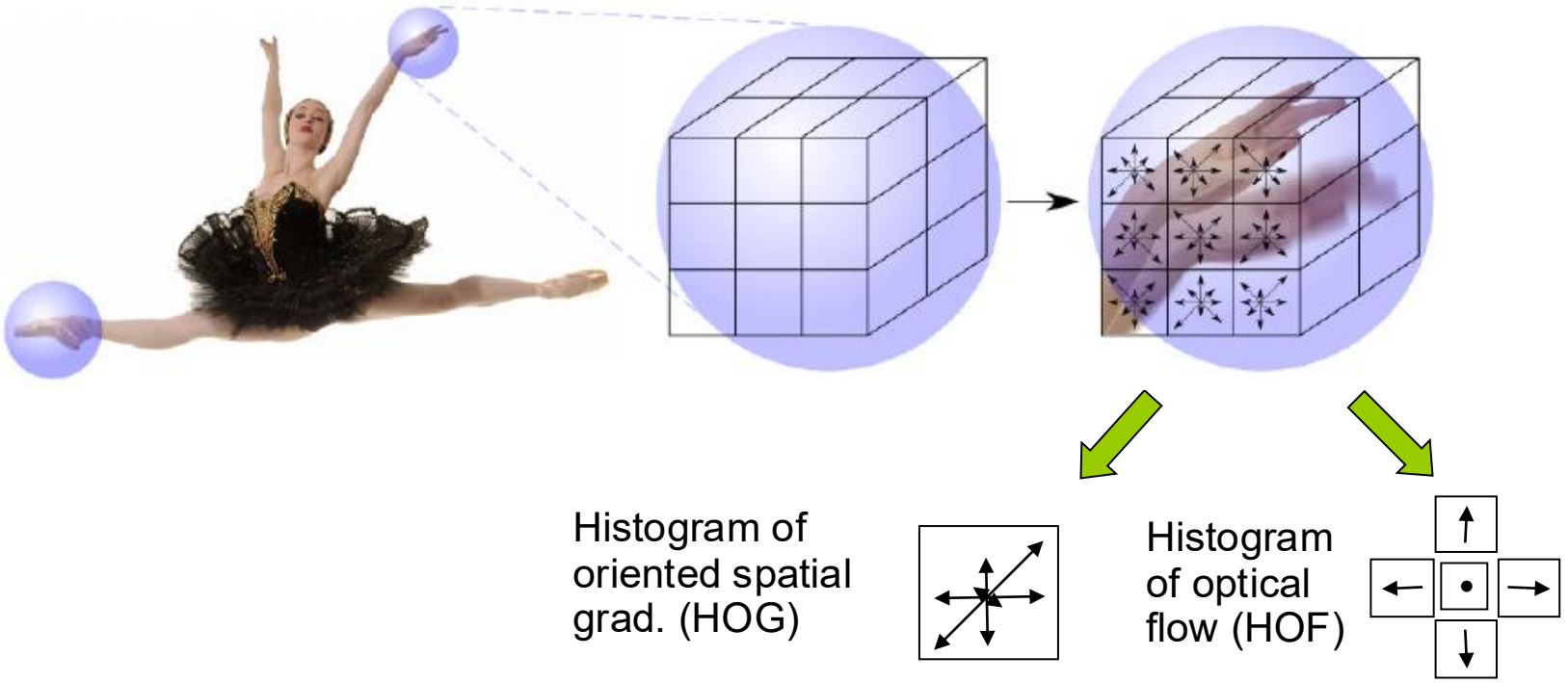
Table 3. Accuracy per action class for the YouTube dataset. We compare with the results reported in [9].



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.

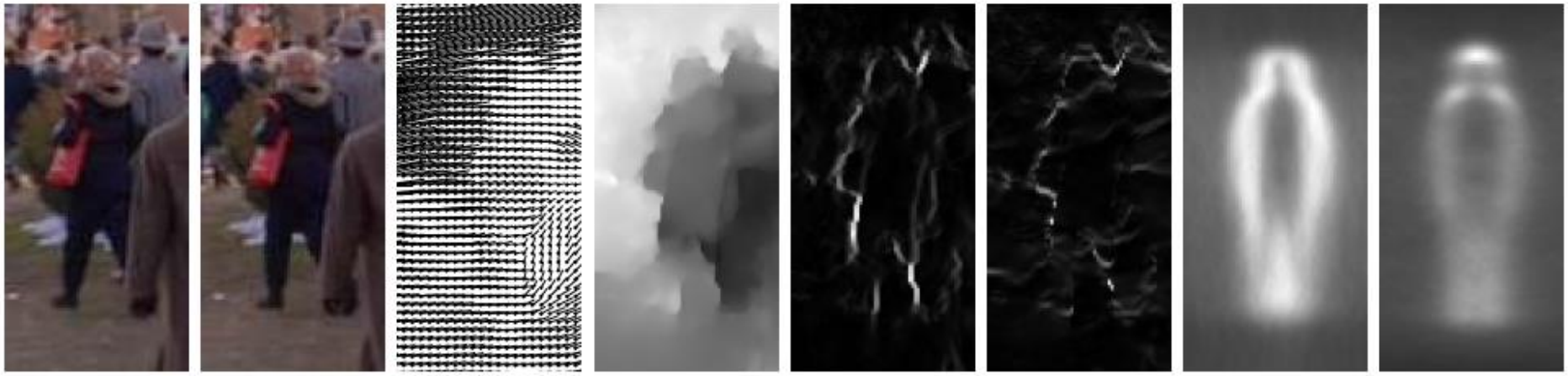
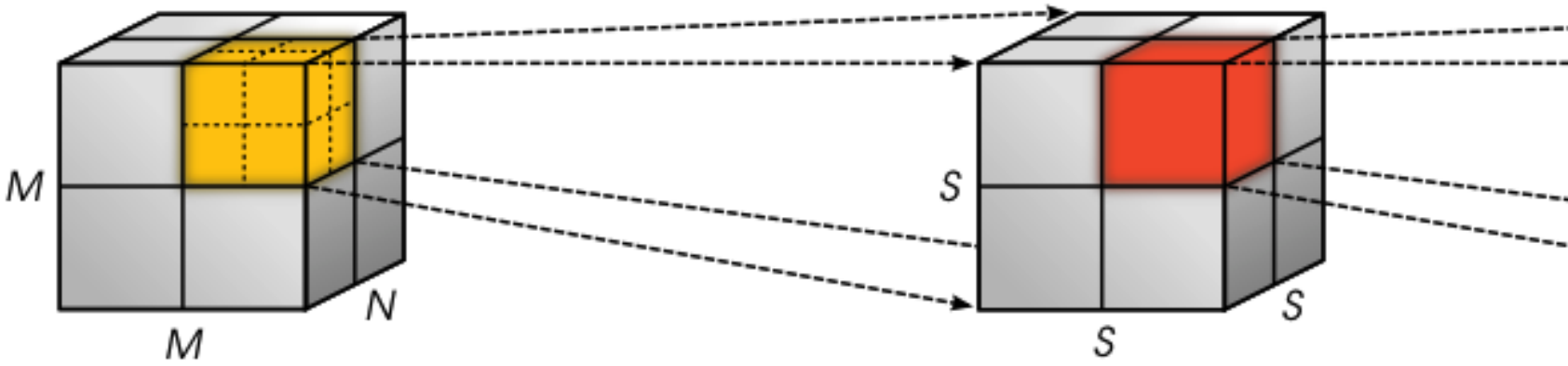


Fig. 3. Illustration of the MBH descriptor. (a,b) Reference images at time t and $t+1$. (c,d) Computed optical flow, and flow magnitude showing motion boundaries. (e,f) Gradient magnitude of flow field $\mathcal{I}^x$, $\mathcal{I}^y$ for image pair (a,b). (g,h) Average MBH descriptor over all training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

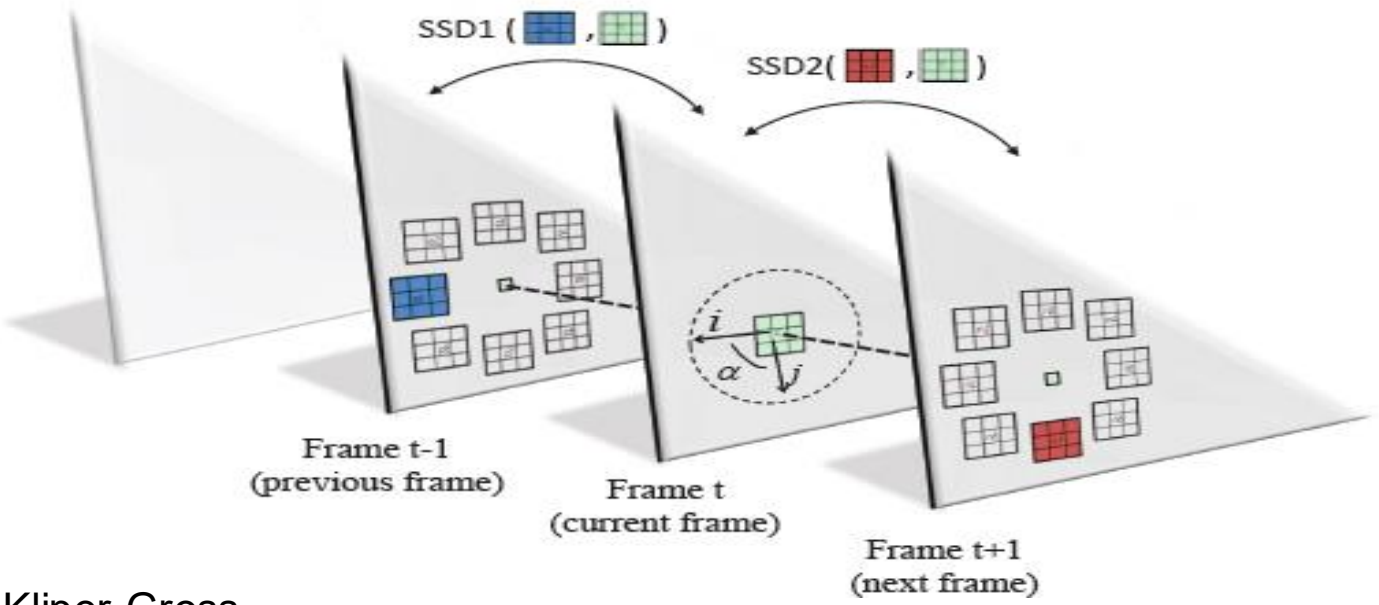
(iv) Gradient computation

- ✔ 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.

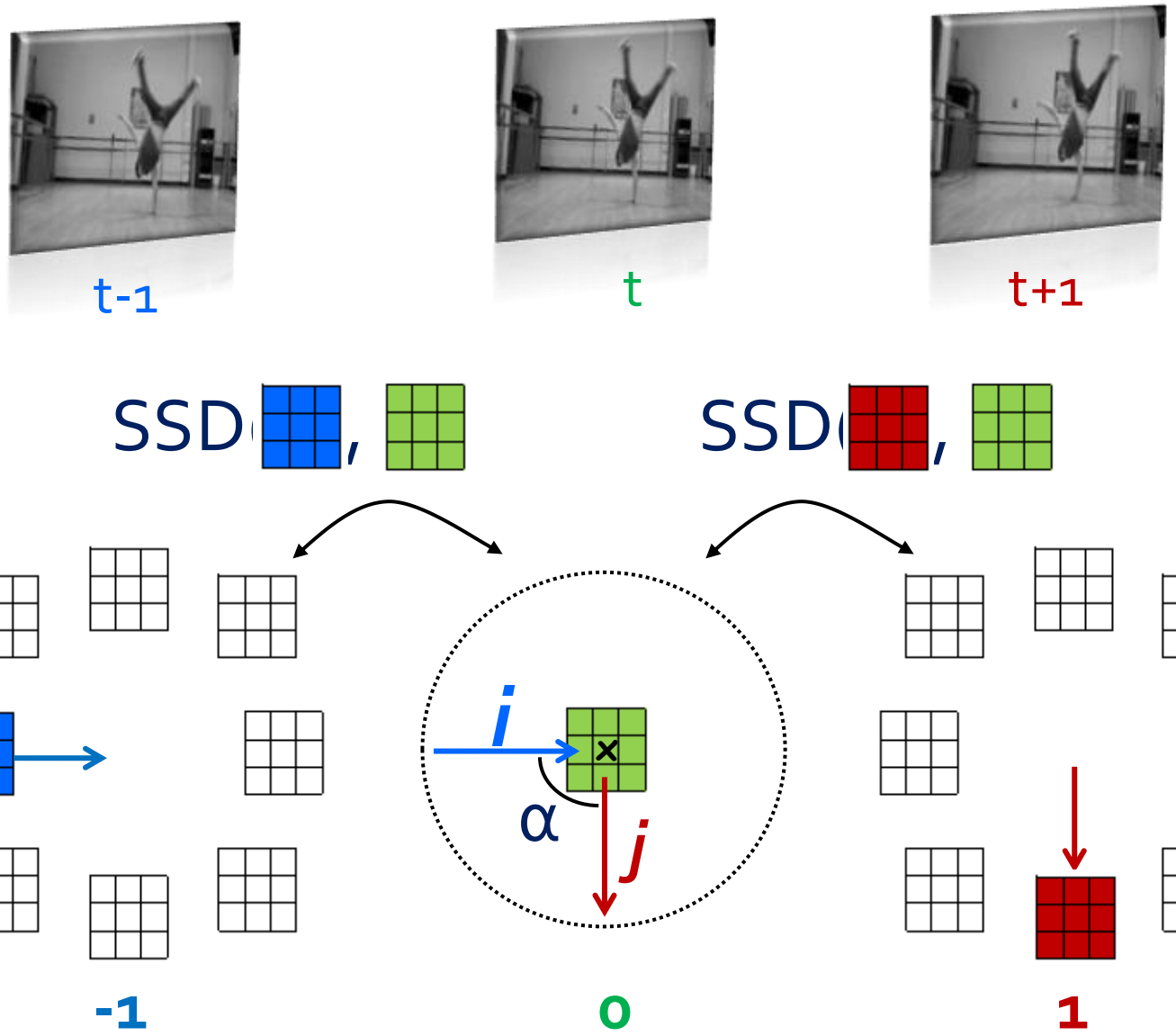
- Local-binary patterns based video descriptor.
  - Dense characterization of motion changes.
  - Captures the shape of moving edges.
  - Methodology incorporates a stabilization mechanism.
- Incorporates a per-pixel encoding using binary/trinary digits.
- Descriptor is frequency of binary/trinary strings.



Slide from O. Kliper-Gross.

Local Descriptors: Motion Interchange Patterns

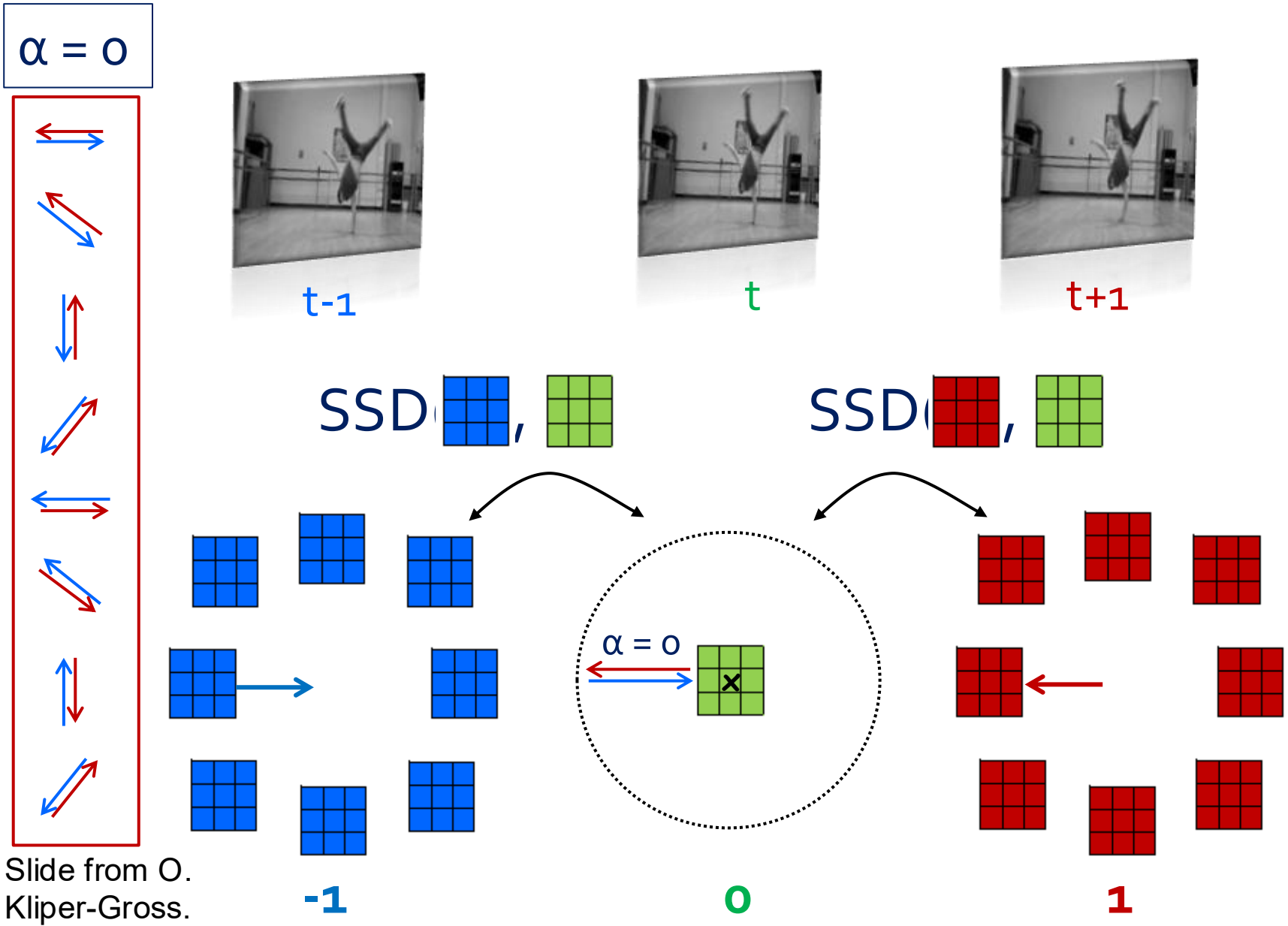
Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.



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.

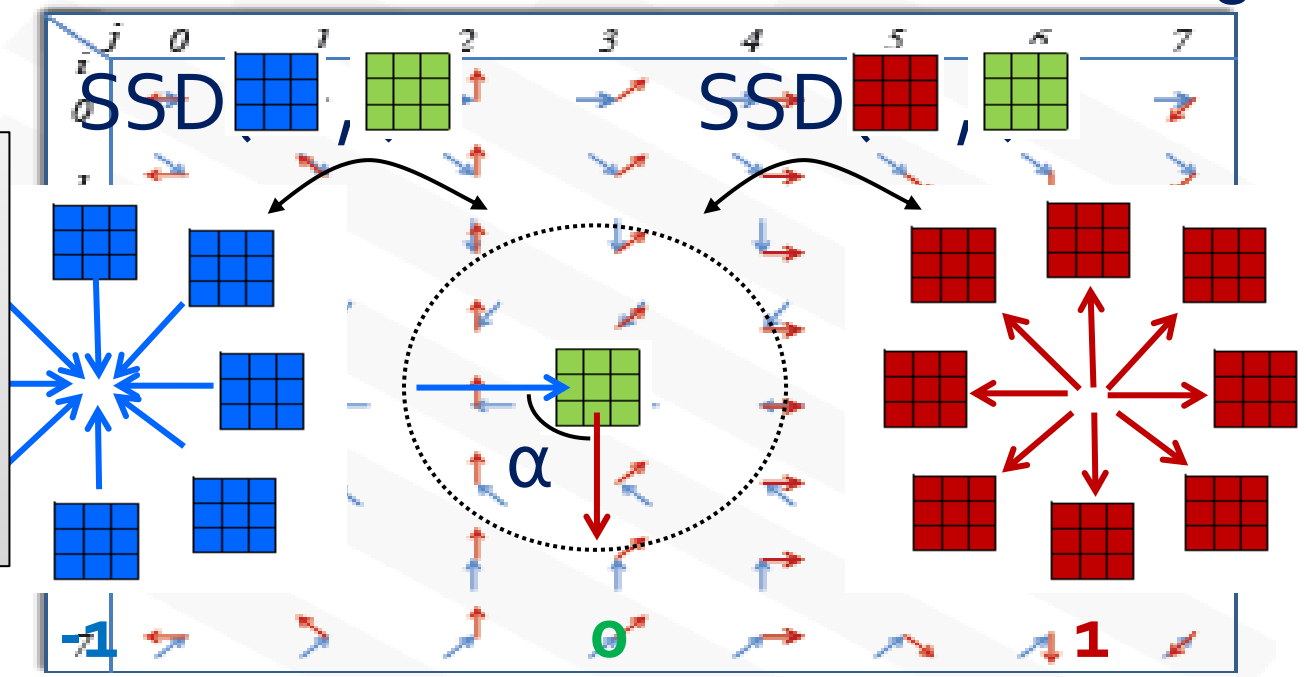


Slide from O. Kliper-Gross.

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.

different α = different channels = diagonals



64-digits trinary code

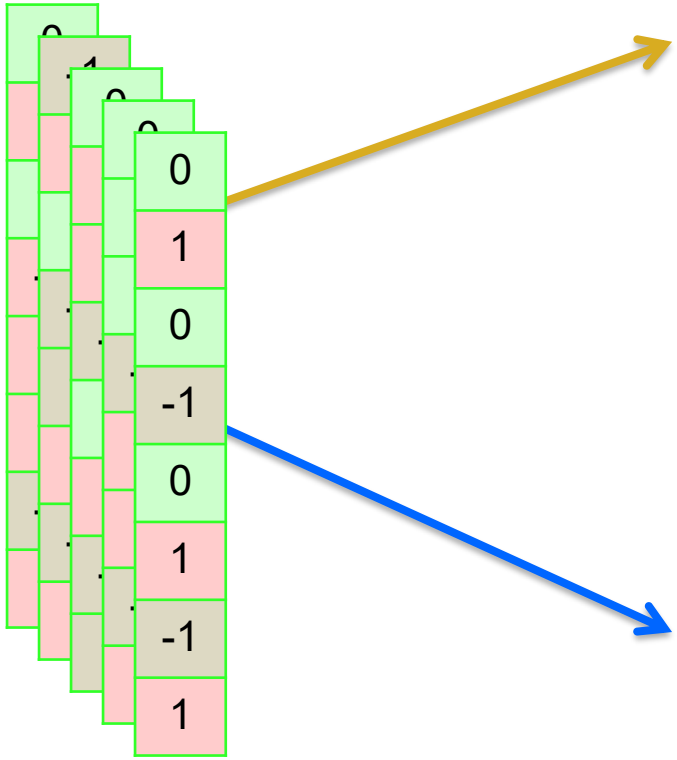
Slide from O. Kliper-Gross.

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.

Each α defines a channel → *8 channels*

Per-pixel 64-digits trinary code



0-255 integer

2 integers per-pixel *Per Channel*

0-255 integer

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

Slide from O. Kliper-Gross.

MIP captures:

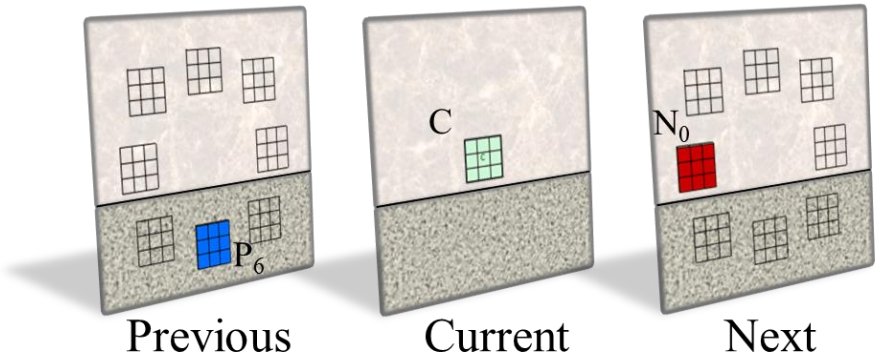
Motion, Motion Changes, and Shape

Local Descriptors: Motion Interchange Patterns

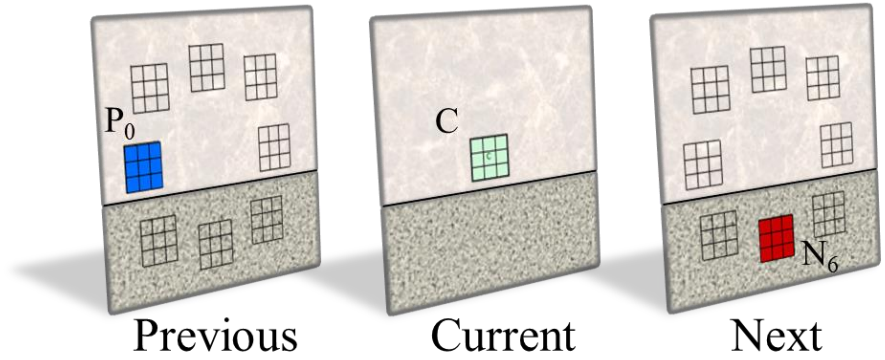
Source: Kliper-Gross et al. "Motion Interchange Patterns for Action Recognition in Unconstrained Videos." ECCV 2012. And the provided slides.

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 ←

Switched patches voted up →

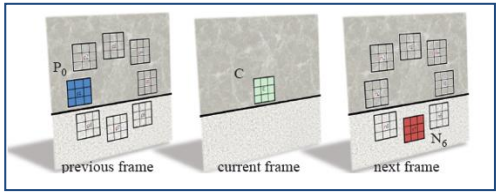
Suppress the code

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 of MIP suppression.



Without Suppression | Original | With Suppression

Slide from O. Kliper-Gross.