Lecture 06
Features

2024-09-18

Sébastien Valade

VNIVERSIDAD NACIONAL AVTONOMA DE MEXICO

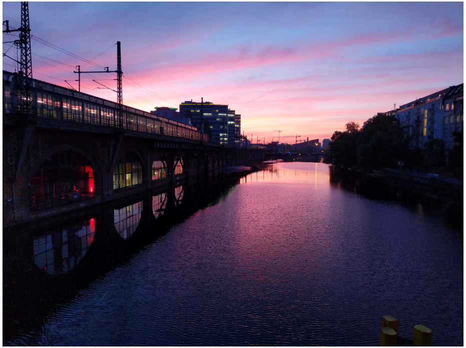
1. Introduction

2. Feature detection

Introduction

Consider two scenes taken from the same location, with slightly different angles and at different times:

Jannowitzbrücke, Berlin



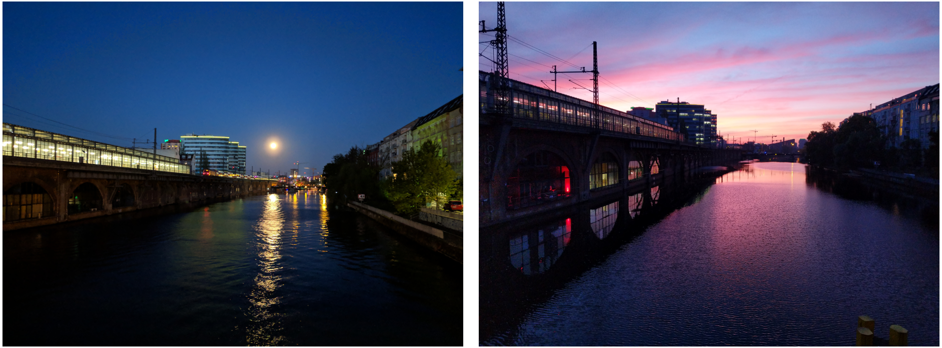
→ small changes in the real world scene lead to big changes in pixel space!

⇒ find **image locations** where we can reliably find correspondences amongst images (regardless of image illumination, camera position, etc.): **detect features**!

Introduction

Consider two scenes taken from the same location, with slightly different angles and at different times:

Jannowitzbrücke, Berlin



→ small changes in the real world scene lead to big changes in pixel space!

⇒ find **image locations** where we can reliably find correspondences amongst images (regardless of image illumination, camera position, etc.): **detect features**!

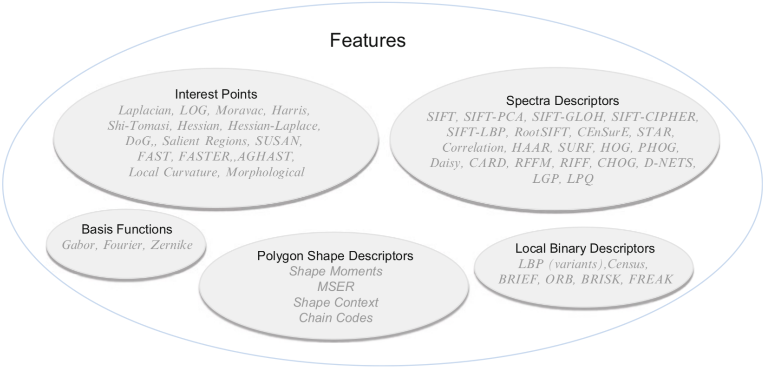
Introduction

Applications

- **Image stitching**: combine multiple images into a single image (i.e., panorama construction $\rightarrow$ last lecture)
- **Motion tracking**: follow objects in a image/video sequence $\rightarrow$ this lecture
- **Structure from Motion**: reconstruct 3D scene from images
- **Object recognition**: recognize objects in an image
- **Image Retrieval**: find similar images in a database
- etc.

Introduction

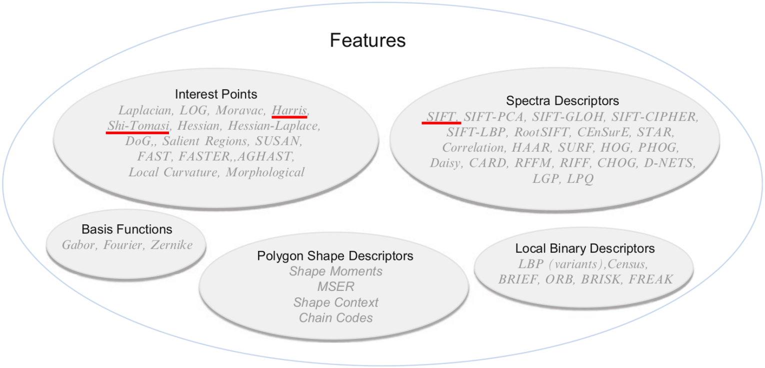
Methods: there are *many* methods and variations in feature description, and terminology can vary across the literature



From: S. Krig, "Computer vision metrics", Springer, 2016

Introduction

Methods: there are *many* methods and variations in feature description, and terminology can vary across the literature



From: S. Krig, "Computer vision metrics", Springer, 2016

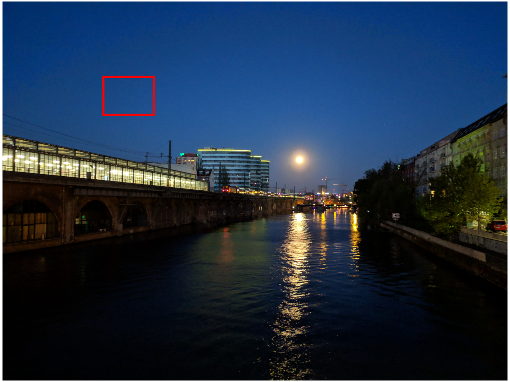
* discussed in this lecture

Which parts of the image are most descriptive? (i.e., what are the requirements for a *good* feature):

- **Invariance** to geometric and radiometric distortions *(e.g., rotation, translation, scaling, illumination)*
- **Salient** compared with surrounding *(i.e. recognizable compared to the neighboring area)*
- **Rareness**: good differentiation

Which parts of the image are most descriptive? (i.e., what are the requirements for a *good* feature):

- **Invariance** to geometric and radiometric distortions *(e.g., rotation, translation, scaling, illumination)*
- **Salient** compared with surrounding *(i.e. recognizable compared to the neighboring area)*
- **Rareness**: good differentiation



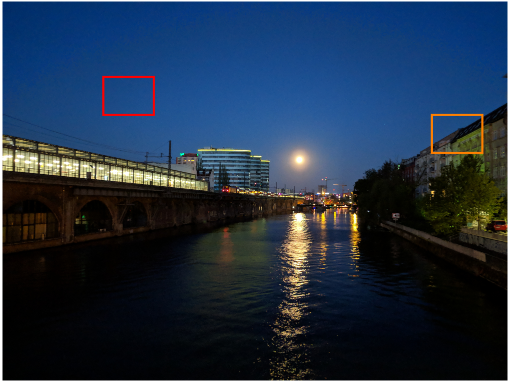
Intuition:

→ regions *without edges*: BAD
nearly impossible to localize

2.1. image region uniqueness

Which parts of the image are most descriptive? (i.e., what are the requirements for a *good* feature):

- **Invariance** to geometric and radiometric distortions *(e.g., rotation, translation, scaling, illumination)*
- **Salient** compared with surrounding *(i.e. recognizable compared to the neighboring area)*
- **Rareness**: good differentiation



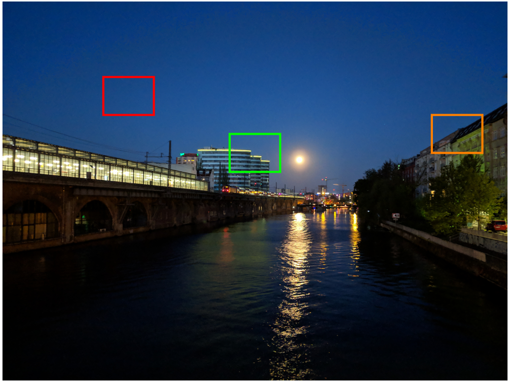
Intuition:

→ regions *without edges*: BAD
nearly impossible to localize

→ regions *with edges*: BETTER
only possible to align the patches along the normal to the edge direction

Which parts of the image are most descriptive? (i.e., what are the requirements for a *good* feature):

- **Invariance** to geometric and radiometric distortions *(e.g., rotation, translation, scaling, illumination)*
- **Salient** compared with surrounding *(i.e. recognizable compared to the neighboring area)*
- **Rareness**: good differentiation



Intuition:

→ regions *without edges*: BAD
nearly impossible to localize

→ regions *with edges*: BETTER
only possible to align the patches along the normal to the edge direction

→ regions *with corner*: GOOD
patches with gradients in different orientations are the easiest to localize

How can we compute the uniqueness of image patches?

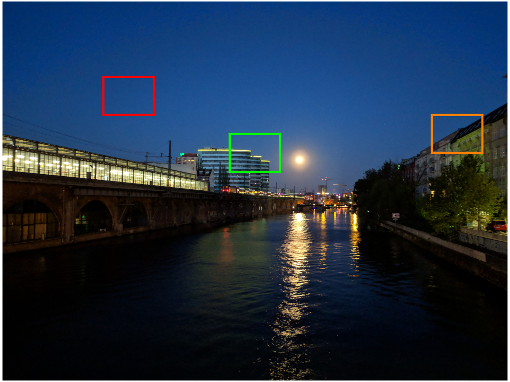
$\Rightarrow$ compute the **self-difference** of the image patch, a.k.a. **autocorrelation function**

$\Rightarrow$ (weighted) summed square difference (SSD) between the patch and itself:

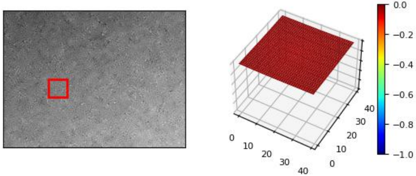
$$E(u, v) = \sum_{x,y} w(x, y) \left[I(x + u, y + v) - I(x, y)\right]^2$$

where $w(x, y)$ = weighting (or window) function (e.g. 1 in window, 0 outside)

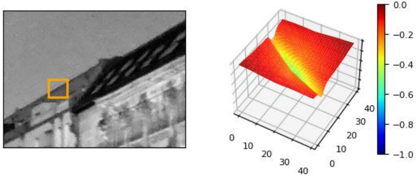
2.1. image region uniqueness



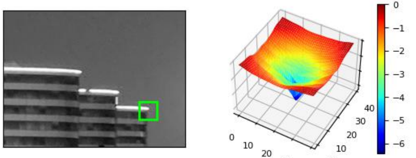
flat ⇒ no stable minimum → cannot be localized



edge ⇒ strong ambiguity along one direction



corner ⇒ strong minimum → can be localized



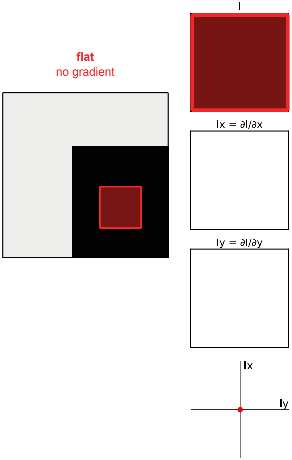
Computing *self-difference* for the whole image is very expensive …

$\Rightarrow$ **need an approximation of the self-difference**: look at nearby **patch gradients I_x and I_y**

2.1. image region uniqueness

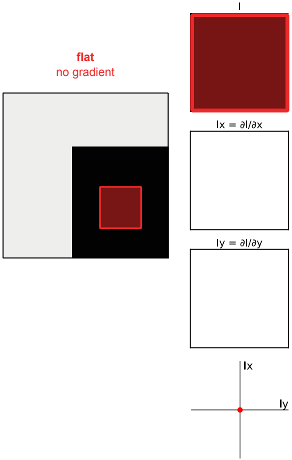
Computing *self-difference* for the whole image is very expensive …

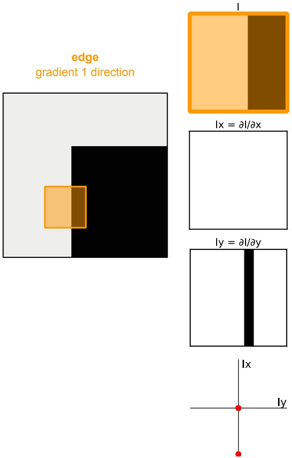
⇒ **need an approximation of the self-difference**: look at nearby **patch gradients I_x and I_y**



Computing *self-difference* for the whole image is very expensive …

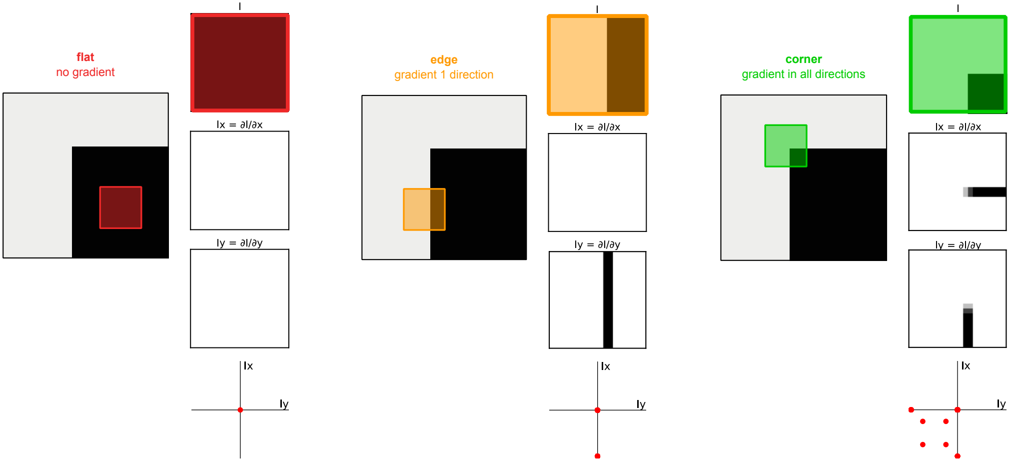
⇒ **need an approximation of the self-difference**: look at nearby **patch gradients I_x and I_y**





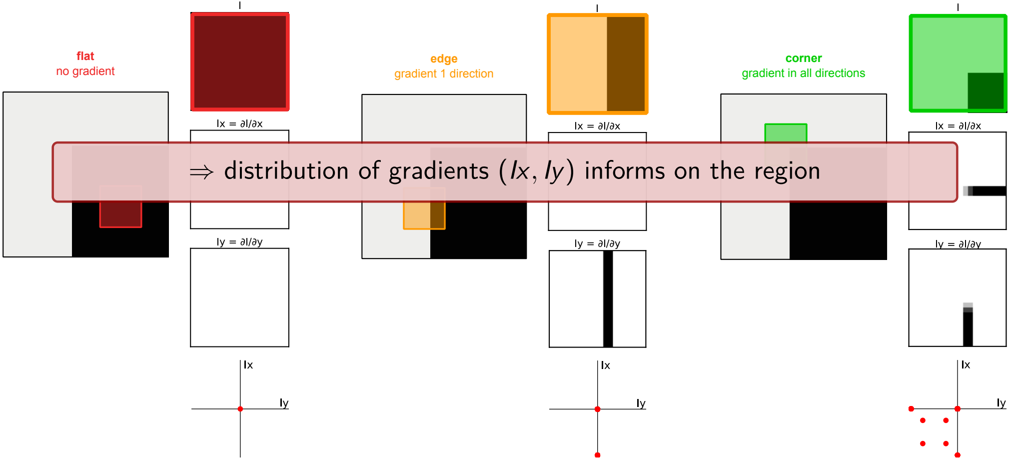
Computing *self-difference* for the whole image is very expensive ...

⇒ **need an approximation of the self-difference**: look at nearby **patch gradients I_x and I_y**



Computing *self-difference* for the whole image is very expensive …

⇒ **need an approximation of the self-difference**: look at nearby **patch gradients I_x and I_y**



⇒ distribution of gradients (Ix, Iy) informs on the region

How can we quantify the orientation and magnitude of the gradients?

1. compute the **covariance matrix** of gradients (Ix, Iy), a.k.a. the *auto-correlation matrix*:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}$$

⇒ using a Taylor Series expansion of the image function I, the auto-correlation function E can be approximated as:

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

$$\simeq \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

How can we quantify the orientation and magnitude of the gradients?

1. compute the **covariance matrix** of gradients (Ix, Iy), a.k.a. the *auto-correlation matrix*:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}$$

$\Rightarrow$ using a Taylor Series expansion of the image function I, the auto-correlation function E can be approximated as:

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

$$\simeq \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

2.2. quantify gradient orientation & magnitude

Math reminders

variance σ^2 = measure of the "spread" or "extent" of the data about some particular axis
= average of the squared differences from the mean
= square of standard deviation (σ)

$$var_x = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N}$$

$$var_y = \frac{\sum_{i=1}^{N}(y_i - \bar{y})^2}{N}$$

covariance = measure the level to which two variables vary together

$$cov_{x,y} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{N-1}$$

covariance matrix $= \begin{bmatrix} var_x & cov_{x,y} \\ cov_{y,x} & var_y \end{bmatrix}$

2.2. quantify gradient orientation & magnitude

Math reminders

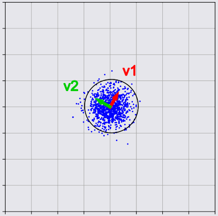
Covariance matrix $= \begin{bmatrix} var_x & cov_{x,y} \\ cov_{y,x} & var_y \end{bmatrix}$

Eigenvalue analysis of covariance matrix $\Rightarrow$ find directions with maximal variance

- **eigenvectors** $(\vec{v}_1, \vec{v}_2)$: represent the directions of the largest variance of the data
- **eigenvalues** (λ_1, λ_2): represent the magnitude of this variance in those directions

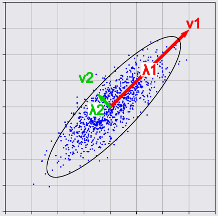
Determinant and trace of covariance matrix

- **determinant** $det(covmat) = \lambda_1\lambda_2$: measures the "spread" of the data captured by the covariance matrix
- **trace** $det(covmat) = \lambda_1 + \lambda_2$: measures the "total variance" captured by the covariance matrix



$$covmat = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

uncorrelated variables



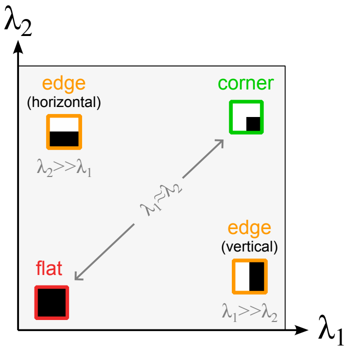
$$covmat = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}$$

highly correlated variables

How can we quantify the orientation and magnitude of the gradients?

2. perform an **eigenvalue analysis** of the auto-correlation matrix M, which produces two **eigenvalues** (λ_1, λ_2) and two **eigenvector** directions $(\vec{v}_1, \vec{v}_2)$

$\Rightarrow$ *the values of the eigenvalues (λ_1, λ_2) will determine the type of region*:



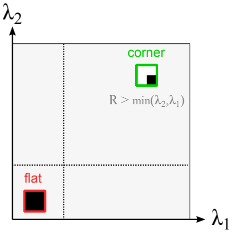
How can we detect corners?

⇒ use a threshold on the eigenvalues (or on a function of the eigenvalues)

Threshold on the eigenvalues:

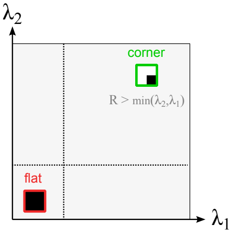
$R = min(\lambda_1, \lambda_2)$

⇒ *Shi – Tomasi corners*



How can we detect corners?

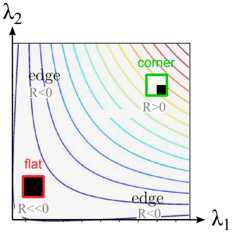
$\Rightarrow$ use a threshold on the eigenvalues (or on a function of the eigenvalues)



Threshold on the eigenvalues:

$$R = min(\lambda_1, \lambda_2)$$

$\Rightarrow$ *Shi – Tomasi corners*



Threshold on a function of the eigenvalues:

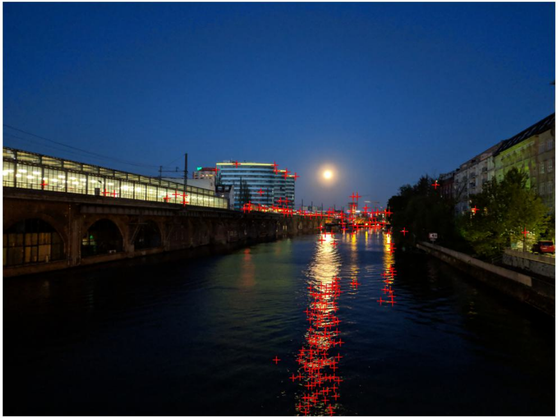
$$R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$$

$$= \det(M) - k(\text{trace}(M))^2$$

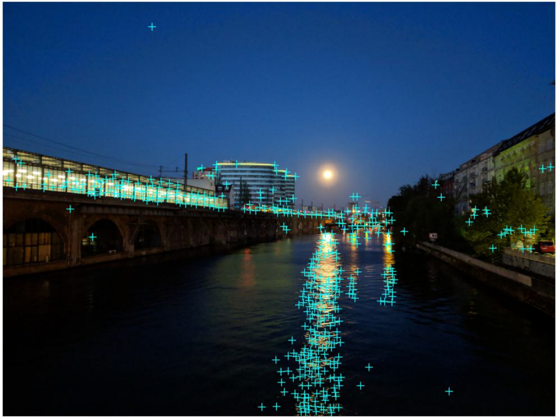
$\Rightarrow$ *Harris corners*

2.3. detect corners

Harris corners

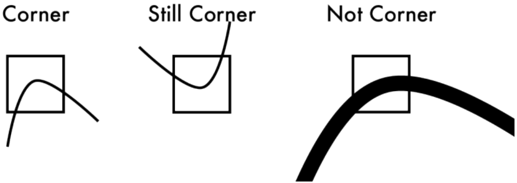


Shi-Tomasi corners



As good as it gets?

⇒ no, corners are invariant to rotation, but not invariant to scale!

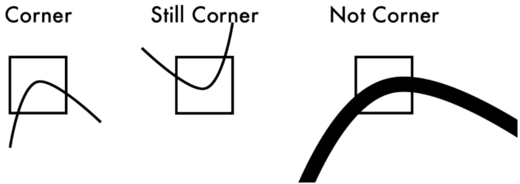


⇒ **SIFT** (Scale-Invariant Feature Transform)!

Lowe (2004), Distinctive Image Features from Scale-Invariant Keypoints, International Journal Of Computer Vision

As good as it gets?

⇒ no, corners are invariant to rotation, but <u>not invariant to scale</u>!



⇒ **SIFT** (Scale-Invariant Feature Transform)!

Lowe (2004), Distinctive Image Features from Scale-Invariant Keypoints, International Journal Of Computer Vision

SIFT

- keypoint detection based on gradients (similar to Harris)
- uses different scales of the image to achieve scale invariance
- uses gradient orientation normalization to achieve scale invariance

Lowe (2004), Distinctive Image Features from Scale-Invariant Keypoints