Seminar: Processing Digital Camera Images
Image Descriptors

Mario Bodemann, Sebastian Koch,
Dmitry Nedospasov, Helen Perkunder

Computer Graphics
Technische Universität Berlin
WS 08/09

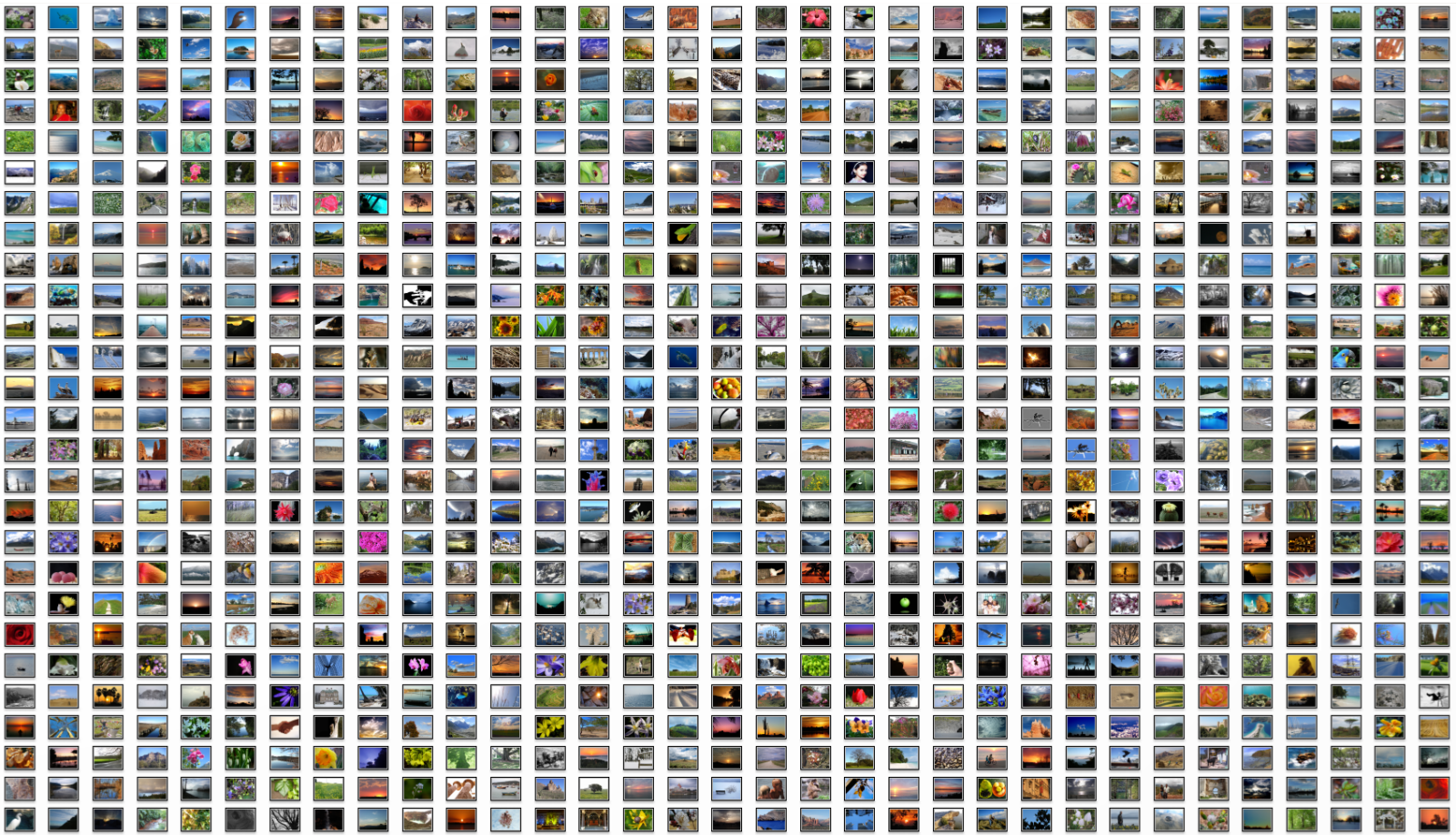
9. Dezember 2008

Image Descriptors

Outline

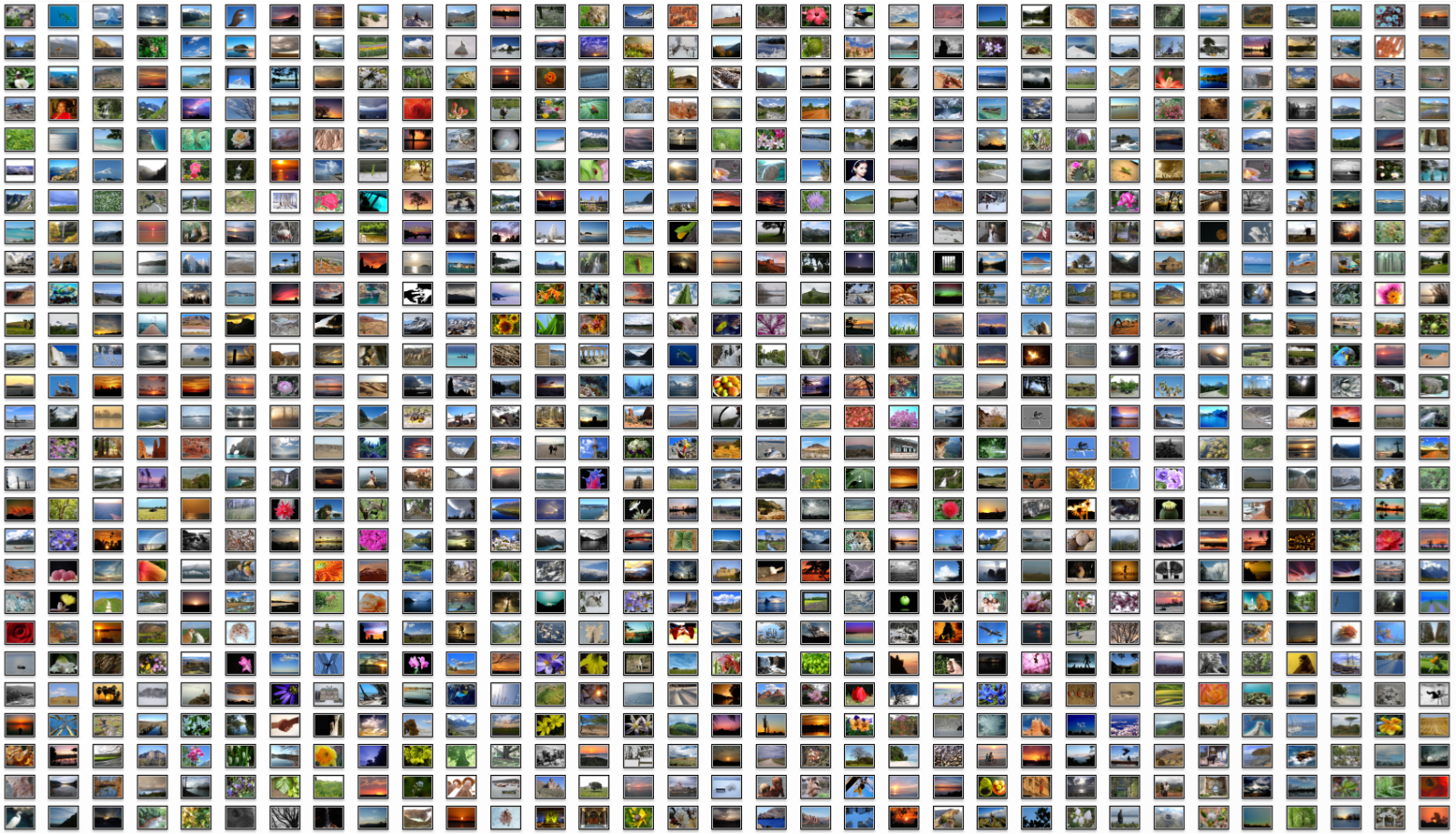
What are we going to be doing?

Searching through and analyzing **millions** of images to create a mosaic image, complete a scene, etc. . .



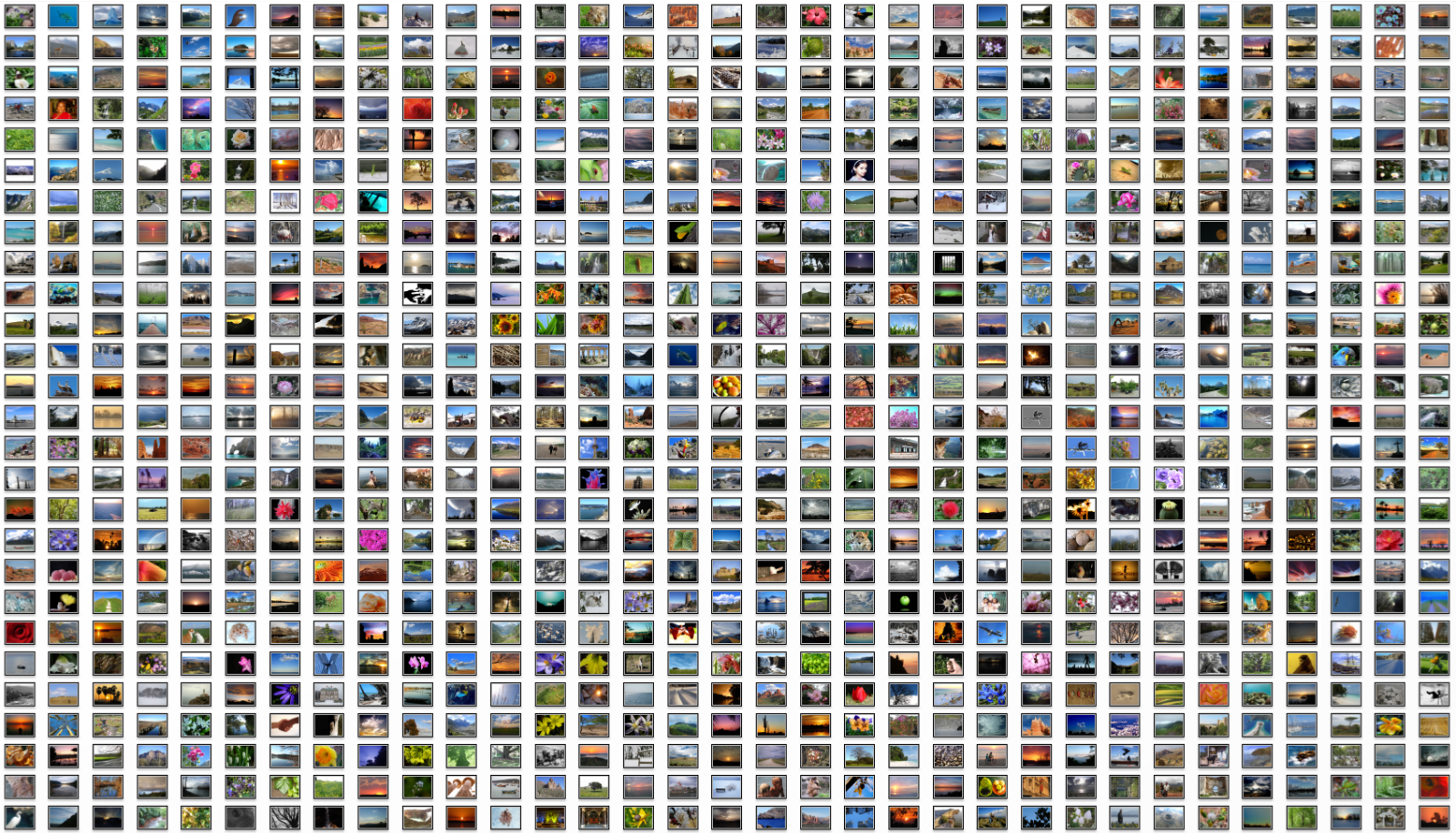
The inherent problems with large image databases

- Long access time to read all the image data
- Long processing time especially with a per-pixel analysis



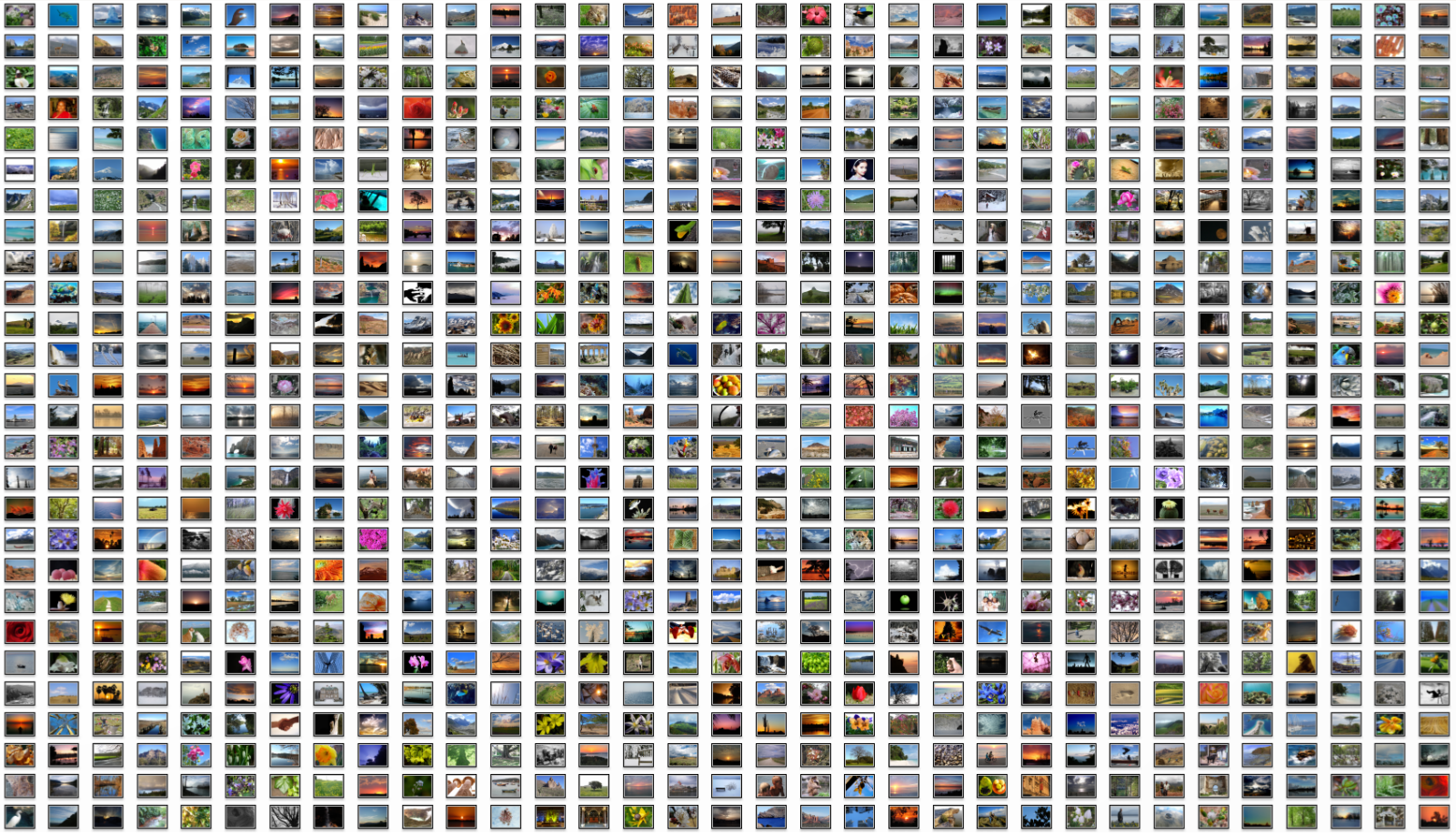
The inherent problems with large image databases

- Long access time to read all the image data
- Long processing time especially with a per-pixel analysis



The inherent problems with large image databases

- Long access time to read all the image data
- Long processing time especially with a per-pixel analysis



What are descriptors and why are they important?

Descriptor - A *piece* of stored information that is used to identify an item in an information storage and retrieval system.[1]

- Descriptors save pertinent information, saving the processing time in future queries
- Descriptors can save image features that are essential for search and comparison

[1] http://wordnet.princeton.edu/perl/webwn?s=descriptor

What are descriptors and why are they important?

Descriptor - A *piece* of stored information that is used to identify an item in an information storage and retrieval system.[1]

- Descriptors save pertinent information, saving the processing time in future queries
- Descriptors can save image features that are essential for search and comparison

[1] http://wordnet.princeton.edu/perl/webwn?s=descriptor

What are descriptors and why are they important?

Descriptor - A *piece* of stored information that is used to identify an item in an information storage and retrieval system.[1]

- Descriptors save pertinent information, saving the processing time in future queries
- Descriptors can save image features that are essential for search and comparison

[1] http://wordnet.princeton.edu/perl/webwn?s=descriptor

How long would it take to read 1TB of data?

- In an ideal situation...
  - $\frac{1 Terabyte}{100 Megabyte/s} = 3$ hours (Internal HD)
  - $\frac{1 Terabyte}{100 Megabits/s} = 23$ hours (Ethernet)
  - $\frac{1 Terabyte}{16 Megabits/s} = 6$ days (DSL-16000)
- The actual sustained reads are lower, especially over a network!

How long would it take to read 1TB of data?

- In an ideal situation...
  - $\frac{1\,Terabyte}{100\,Megabyte/s} = 3$ hours (Internal HD)
  - $\frac{1\,Terabyte}{100\,Megabits/s} = 23$ hours (Ethernet)
  - $\frac{1\,Terabyte}{16\,Megabits/s} = 6$ days (DSL-16000)
- The actual sustained reads are lower, especially over a network!

How long would it take to read 1TB of data?

- In an ideal situation...
  - $\frac{1\,Terabyte}{100\,Megabyte/s} = 3$ hours (Internal HD)
  - $\frac{1\,Terabyte}{100\,Megabits/s} = 23$ hours (Ethernet)
  - $\frac{1\,Terabyte}{16\,Megabits/s} = 6$ days (DSL-16000)
- The actual sustained reads are lower, especially over a network!

How long would it take to read 1TB of data?

- In an ideal situation...
  - $\frac{1\,Terabyte}{100\,Megabyte/s} = 3$ hours (Internal HD)
  - $\frac{1\,Terabyte}{100\,Megabits/s} = 23$ hours (Ethernet)
  - $\frac{1\,Terabyte}{16\,Megabits/s} = 6$ days (DSL-16000)
- The actual sustained reads are lower, especially over a network!

How long would it take to read 1TB of data?

- In an ideal situation...
  - $\frac{1\,Terabyte}{100\,Megabyte/s} = 3$ hours (Internal HD)
  - $\frac{1\,Terabyte}{100\,Megabits/s} = 23$ hours (Ethernet)
  - $\frac{1\,Terabyte}{16\,Megabits/s} = 6$ days (DSL-16000)
- The actual sustained reads are lower, especially over a network!

How long would it take to read 1TB of data?

- In an ideal situation...
  - $\frac{1\,Terabyte}{100\,Megabyte/s} = 3$ hours (Internal HD)
  - $\frac{1\,Terabyte}{100\,Megabits/s} = 23$ hours (Ethernet)
  - $\frac{1\,Terabyte}{16\,Megabits/s} = 6$ days (DSL-16000)
- The actual sustained reads are lower, especially over a network!

One way to do it...

1. generate small data-structures (descriptors) from the original images
   - *faster data access*
   - *reduced data (could be stored locally)*
2. save the *interesting features* in the descriptor
   - *original would have to be transformed first*
   - *reduce future processing time*
3. result: descriptor that can be compared with other descriptors
   - *calculate various distances (Euclidean, etc.)*
   - *traverse the original database one time!*

One way to do it...

1. generate small data-structures (descriptors) from the original images
   - *faster data access*
   - *reduced data (could be stored locally)*
2. save the *interesting features* in the descriptor
   - *original would have to be transformed first*
   - *reduce future processing time*
3. result: descriptor that can be compared with other descriptors
   - *calculate various distances (Euclidean, etc.)*
   - *traverse the original database one time!*

One way to do it...

1. generate small data-structures (descriptors) from the original images
   - *faster data access*
   - *reduced data (could be stored locally)*
2. save the *interesting features* in the descriptor
   - *original would have to be transformed first*
   - *reduce future processing time*
3. result: descriptor that can be compared with other descriptors
   - *calculate various distances (Euclidean, etc.)*
   - *traverse the original database one time!*

One way to do it...

1. generate small data-structures (descriptors) from the original images
   - *faster data access*
   - *reduced data (could be stored locally)*
2. save the *interesting features* in the descriptor
   - *original would have to be transformed first*
   - *reduce future processing time*
3. result: descriptor that can be compared with other descriptors
   - *calculate various distances (Euclidean, etc.)*
   - *traverse the original database one time!*

One way to do it...

1. generate small data-structures (descriptors) from the original images
   - *faster data access*
   - *reduced data (could be stored locally)*
2. save the *interesting features* in the descriptor
   - *original would have to be transformed first*
   - *reduce future processing time*
3. result: descriptor that can be compared with other descriptors
   - *calculate various distances (Euclidean, etc.)*
   - *traverse the original database one time!*

One way to do it...

1. generate small data-structures (descriptors) from the original images
   - *faster data access*
   - *reduced data (could be stored locally)*
2. save the *interesting features* in the descriptor
   - *original would have to be transformed first*
   - *reduce future processing time*
3. result: descriptor that can be compared with other descriptors
   - *calculate various distances (Euclidean, etc.)*
   - *traverse the original database one time!*

One way to do it...

1. generate small data-structures (descriptors) from the original images
   - *faster data access*
   - *reduced data (could be stored locally)*
2. save the *interesting features* in the descriptor
   - *original would have to be transformed first*
   - *reduce future processing time*
3. result: descriptor that can be compared with other descriptors
   - *calculate various distances (Euclidean, etc.)*
   - *traverse the original database one time!*

What kinds of image descriptors are there?

- Meta-Information based: EXIF-Data, (Geo-)Tags. . . but are not interesting for this course
- Image based:
  - Pixel-based: Tiny-Images, Color Histograms, Texton Histograms
  - Structure-based: Fourier, Gradient, Geometric, Gist

What kinds of image descriptors are there?

- Meta-Information based: EXIF-Data, (Geo-)Tags. . . but are not interesting for this course
- Image based:
  - Pixel-based: Tiny-Images, Color Histograms, Texton Histograms
  - Structure-based: Fourier, Gradient, Geometric, Gist

What kinds of image descriptors are there?

- Meta-Information based: EXIF-Data, (Geo-)Tags. . . but are not interesting for this course
- Image based:
  - Pixel-based: Tiny-Images, Color Histograms, Texton Histograms
  - Structure-based: Fourier, Gradient, Geometric, Gist

What kinds of image descriptors are there?

- Meta-Information based: EXIF-Data, (Geo-)Tags. . . but are not interesting for this course
- Image based:
  - Pixel-based: Tiny-Images, Color Histograms, Texton Histograms
  - Structure-based: Fourier, Gradient, Geometric, Gist

What kinds of image descriptors are there?

- Meta-Information based: EXIF-Data, (Geo-)Tags. . . but are not interesting for this course
- Image based:
  - Pixel-based: Tiny-Images, Color Histograms, Texton Histograms
  - Structure-based: Fourier, Gradient, Geometric, Gist

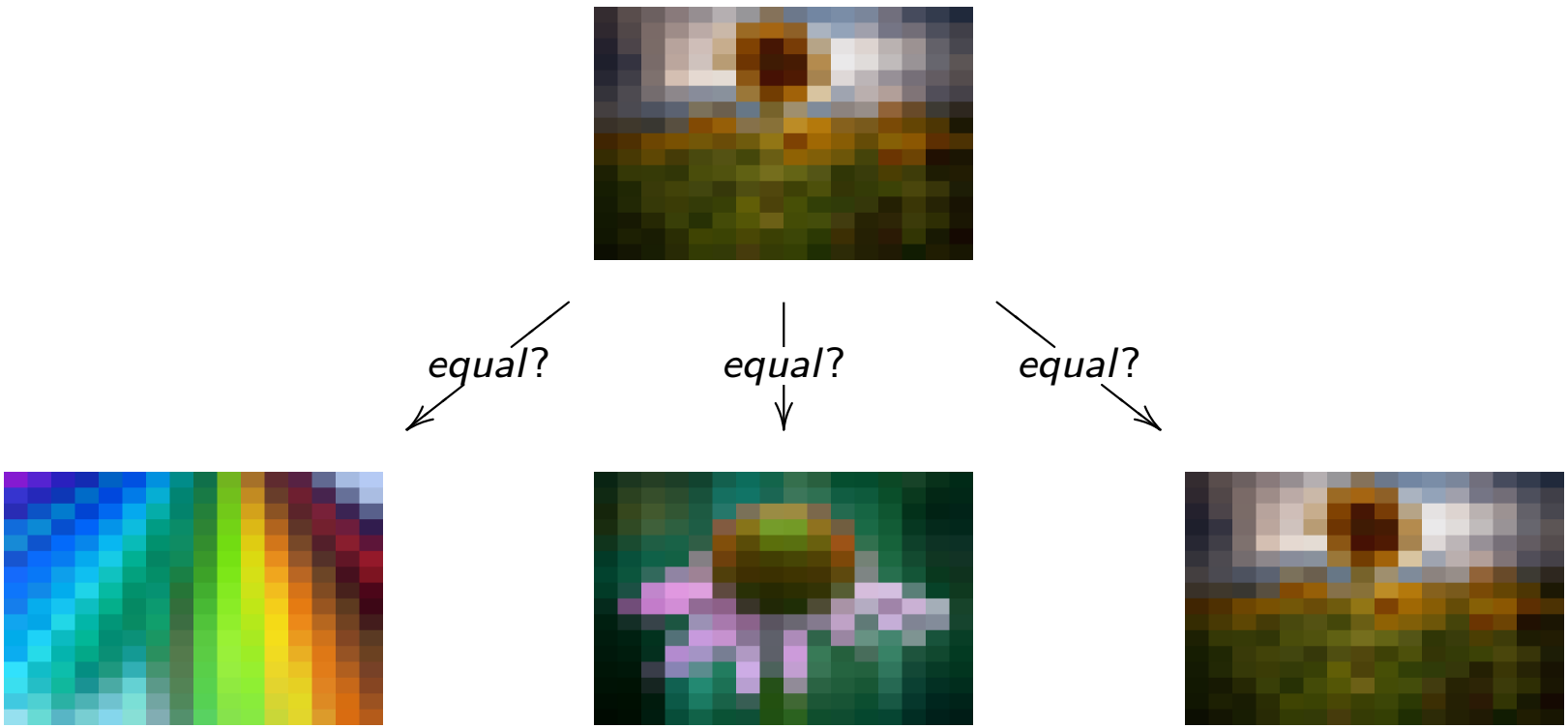
Outline

Tiny Image Source[2]

[2] from http://www.flickr.com/photos/alphageek/2759567956/

Tiny Image Descriptor

Comparison with database

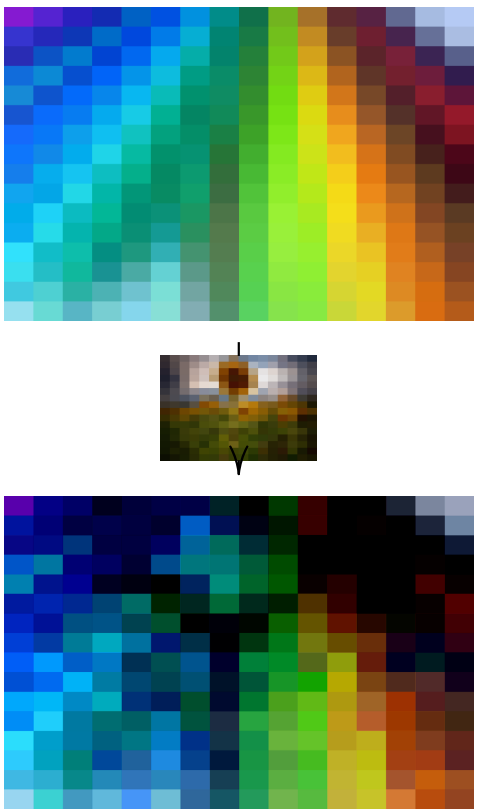


[3] from http://www.flickr.com/photos/40513596@N00/71762740
[4] from http://www.flickr.com/photos/alphageek/2993813155/

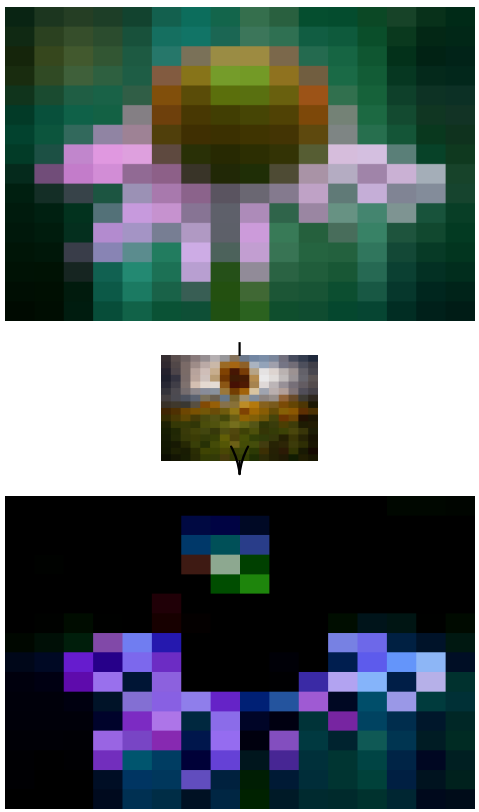
How to compare?

1. Estimate value for each pixel
2. Calculate difference to descriptor (using suitable colorspace)
3. Weigh higher errors by using square of the sum
4. Store the resulting difference
5. Sum up all differences
6. Take the smallest sum for best match

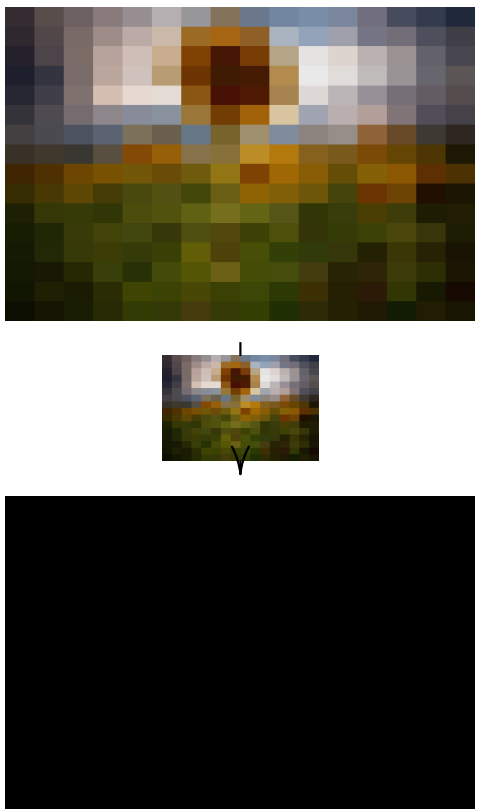
Comparision examples



14862080



437120



0

Tiny Images at a glance

Advantages

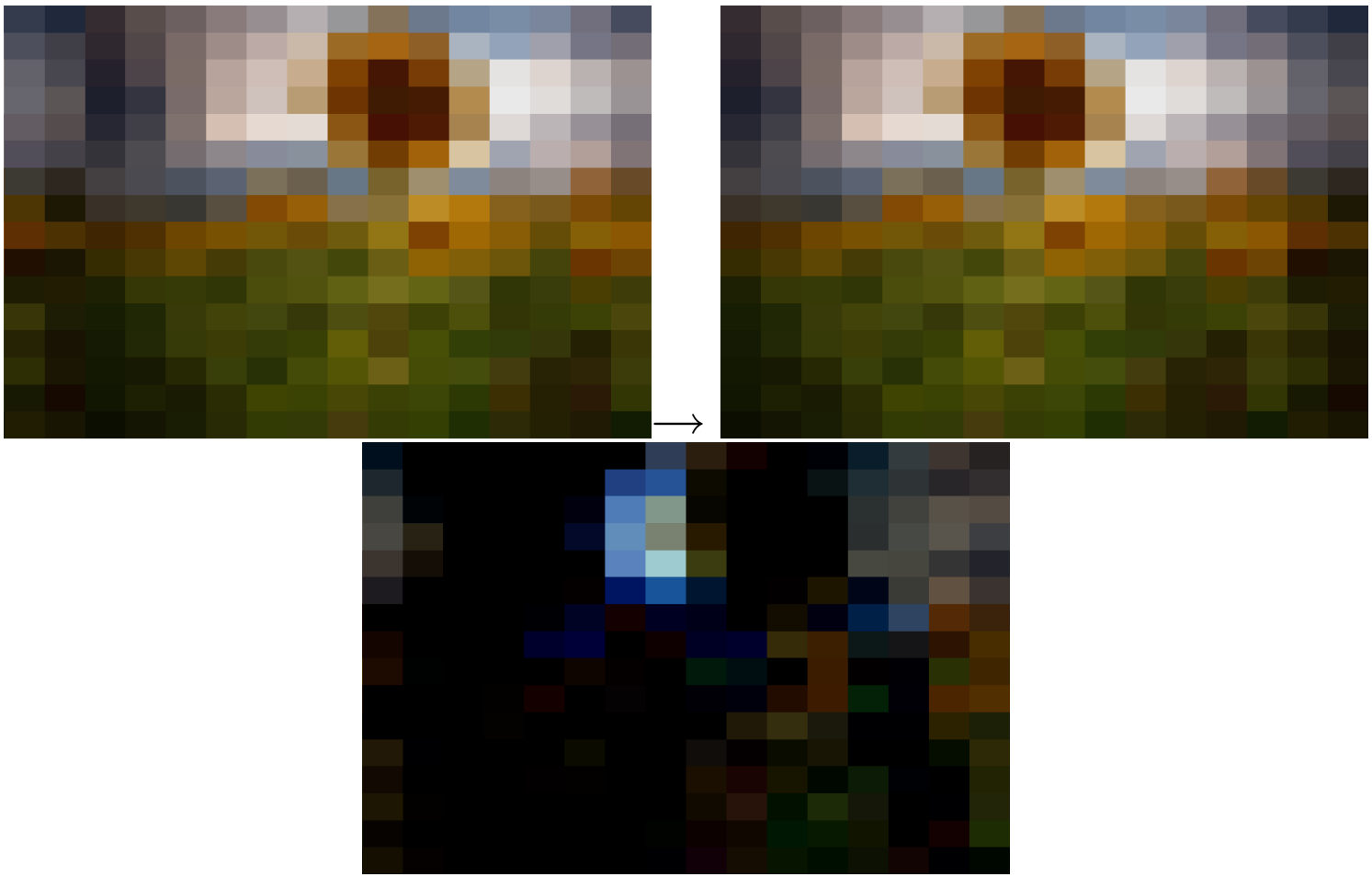
- simple
- good result on large datasets
- location dependent

Disadvantages

- dependent on colorspace
- only useful for color information

Location dependency

- Shifting the descriptor 2 pixel to right (warping)
- completely different result



Outline

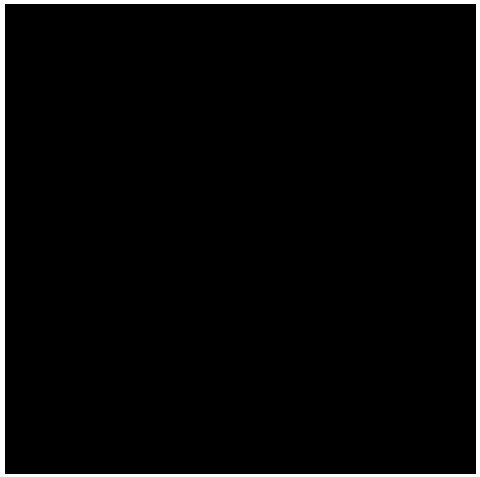
Image Histograms

What are image histograms?

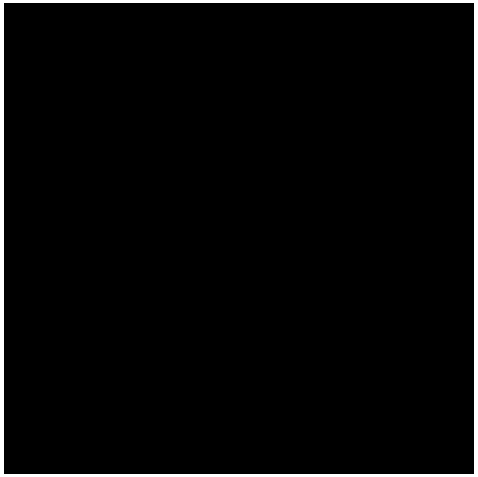
Several Types of Image Histograms

- Color histograms
- Texton histograms
- Line feature histograms
- ...

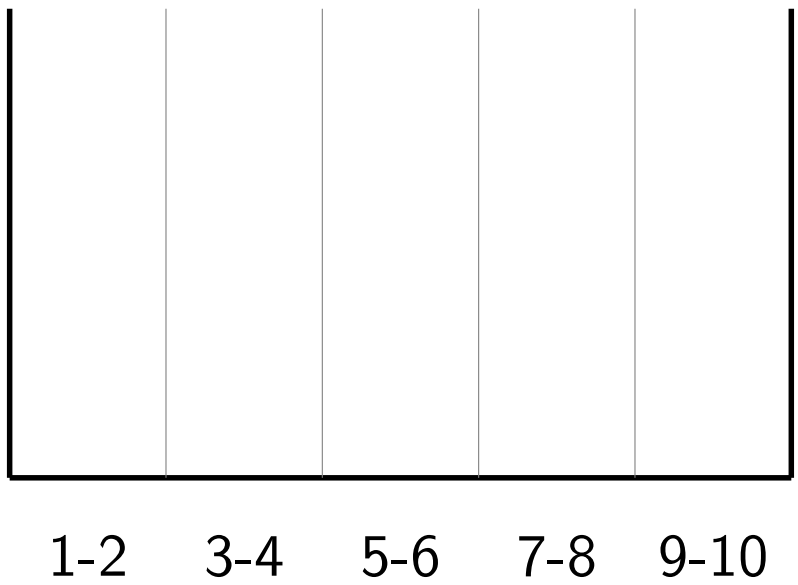
Definition of Color Histograms



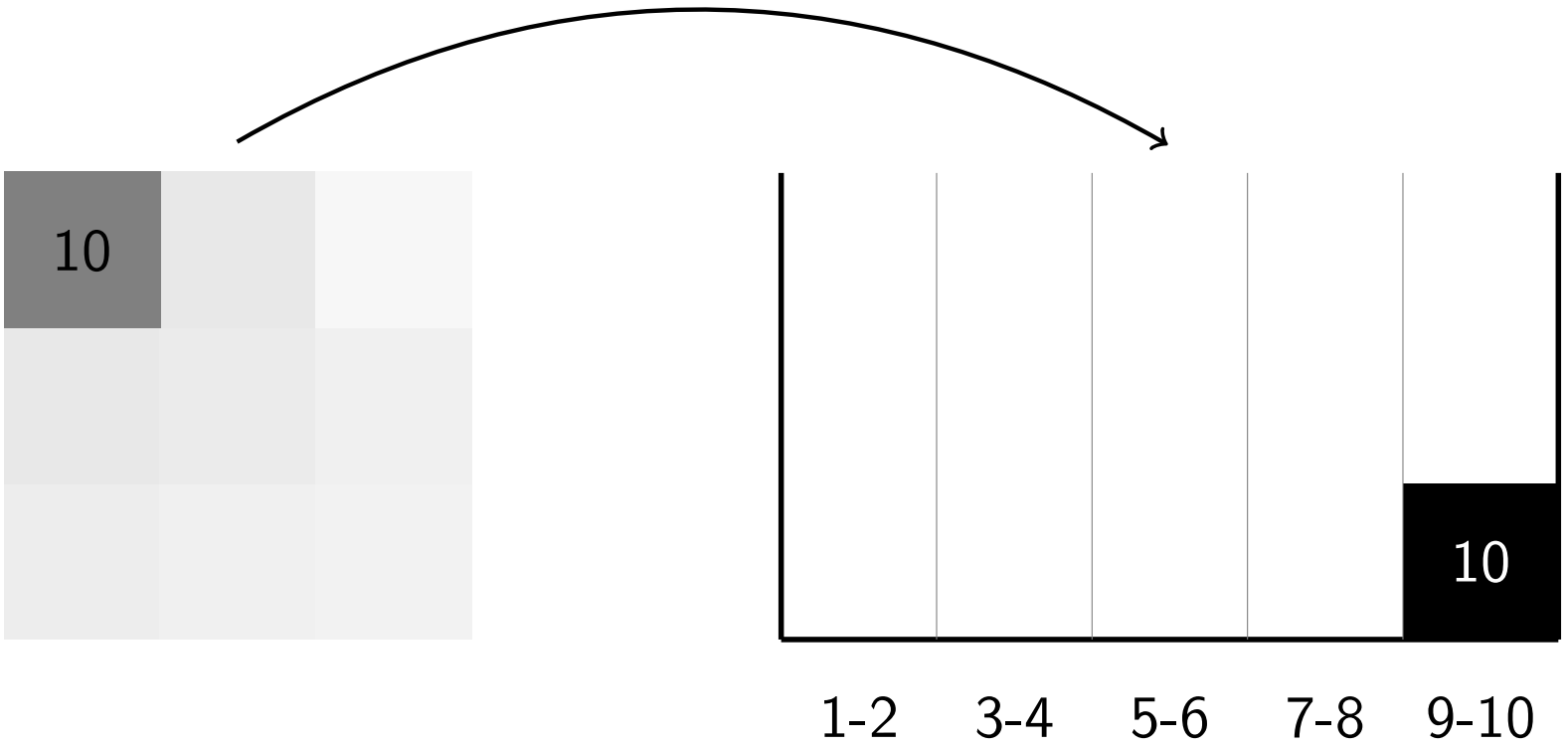
Definition of Color Histograms



Definition of Color Histograms

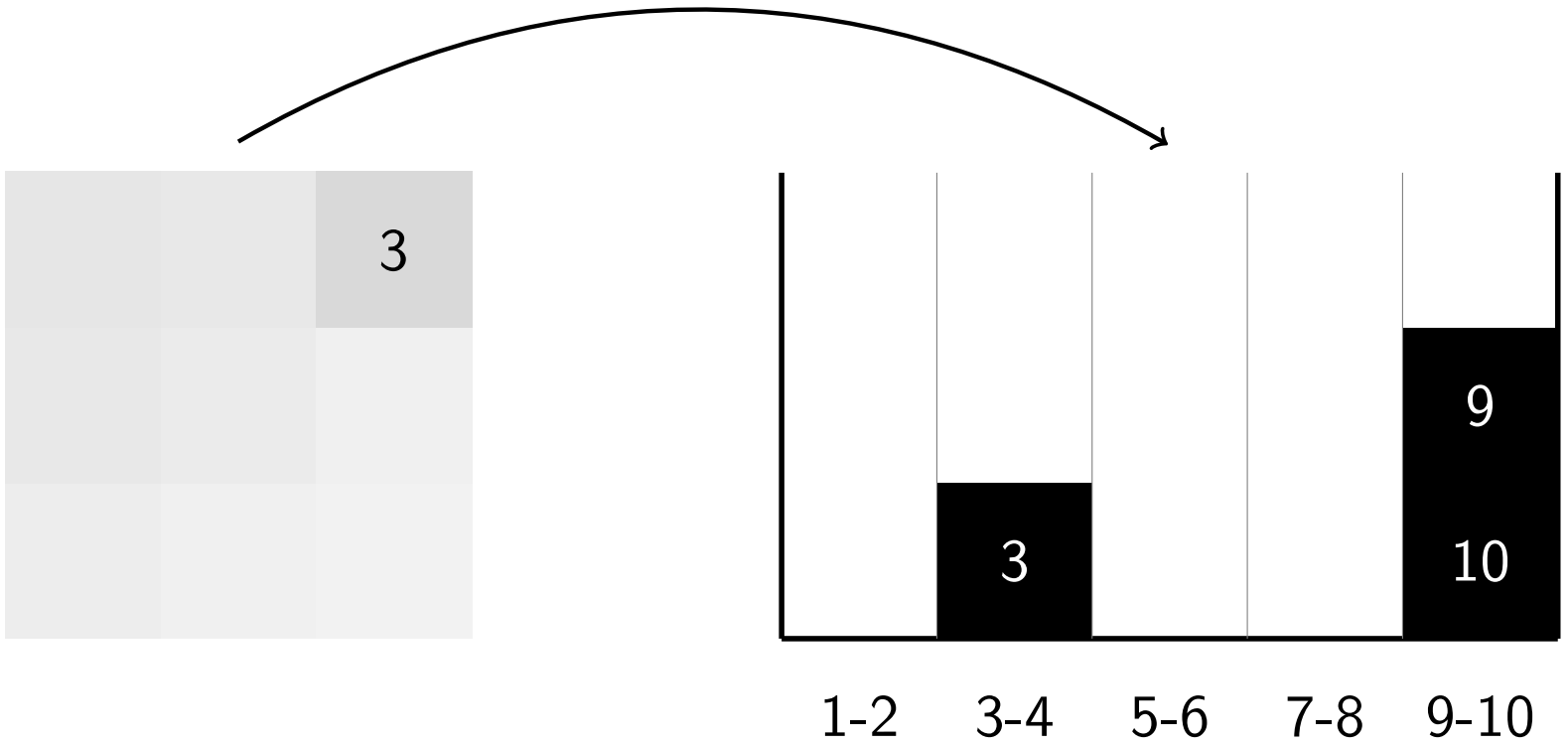


Definition of Color Histograms

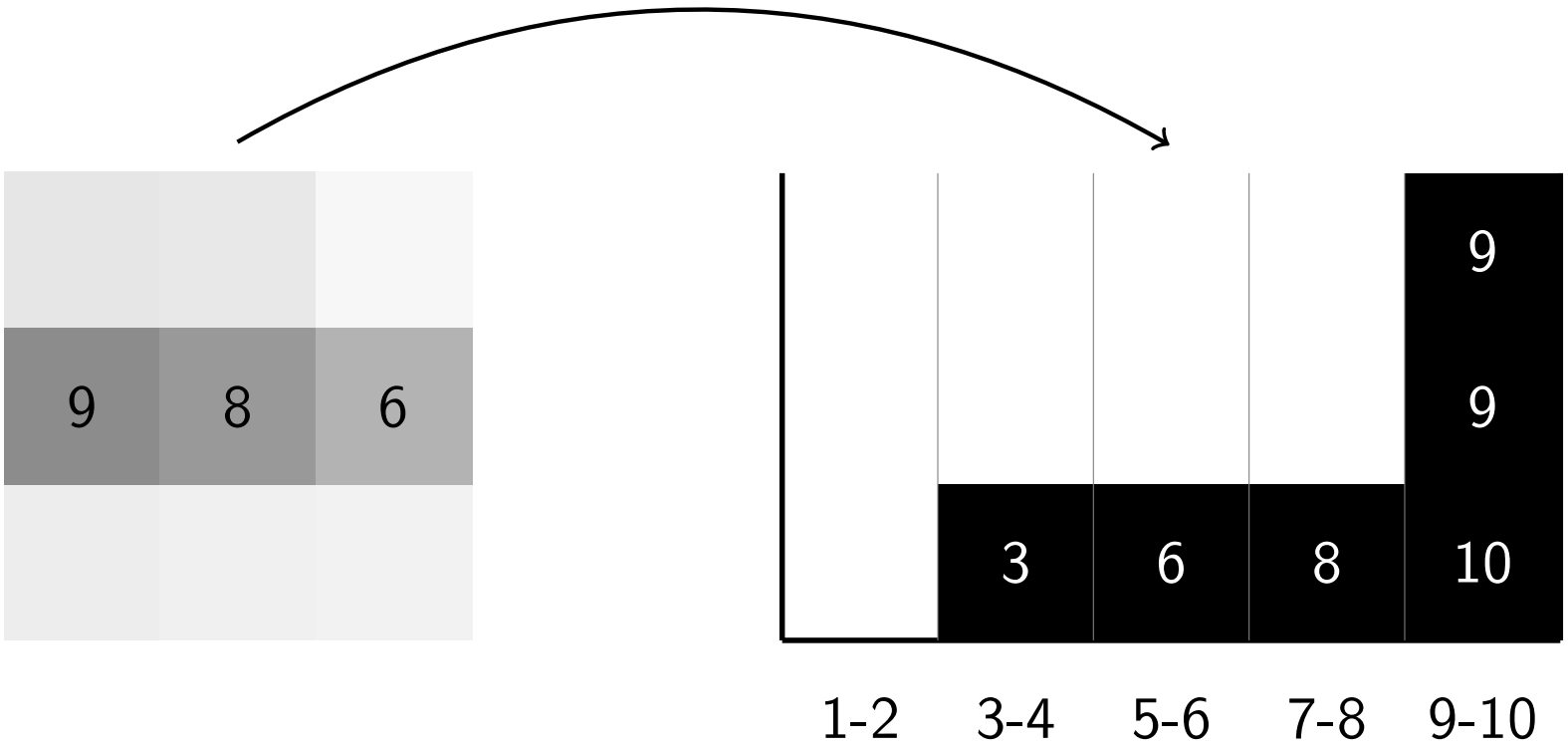


Definition of Color Histograms

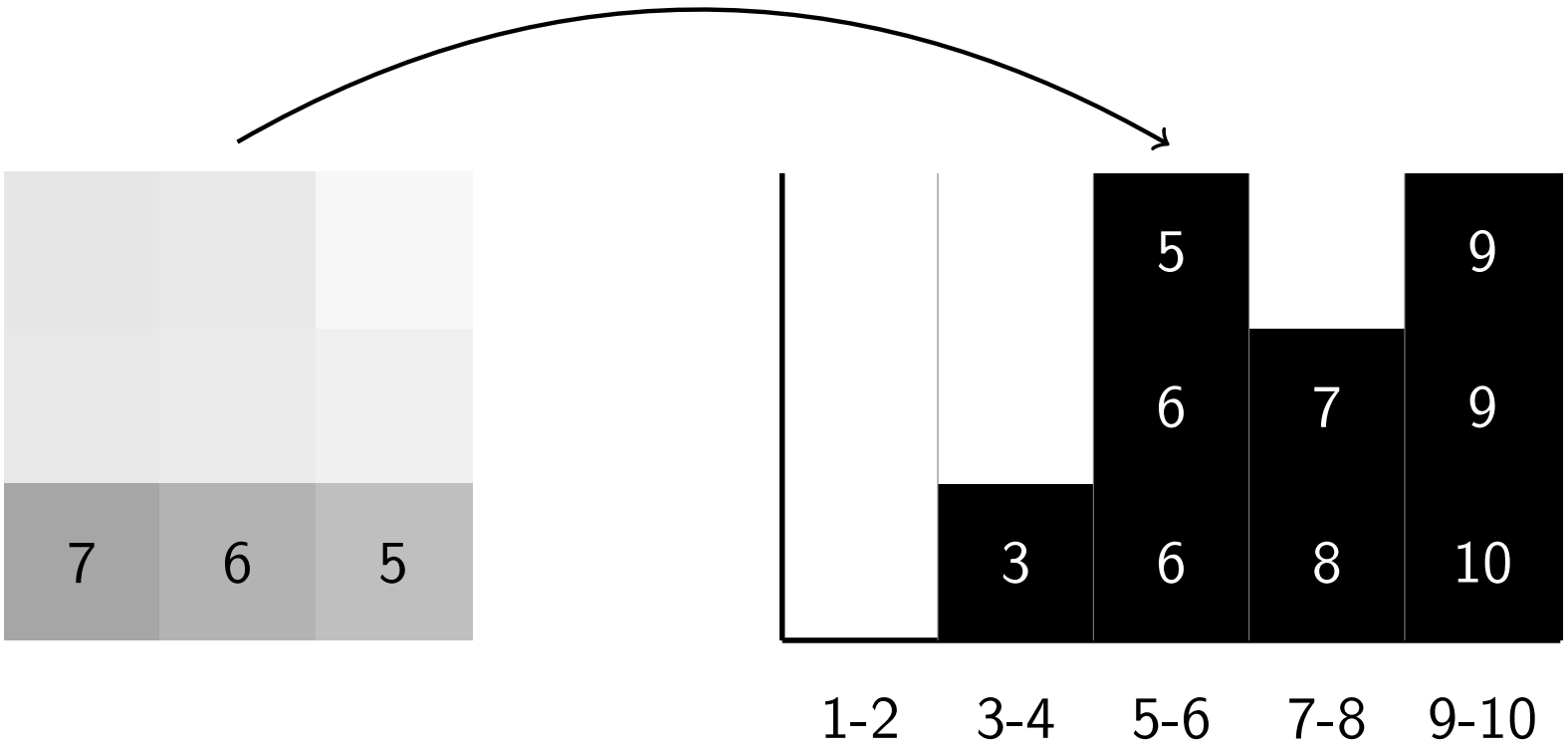
Definition of Color Histograms



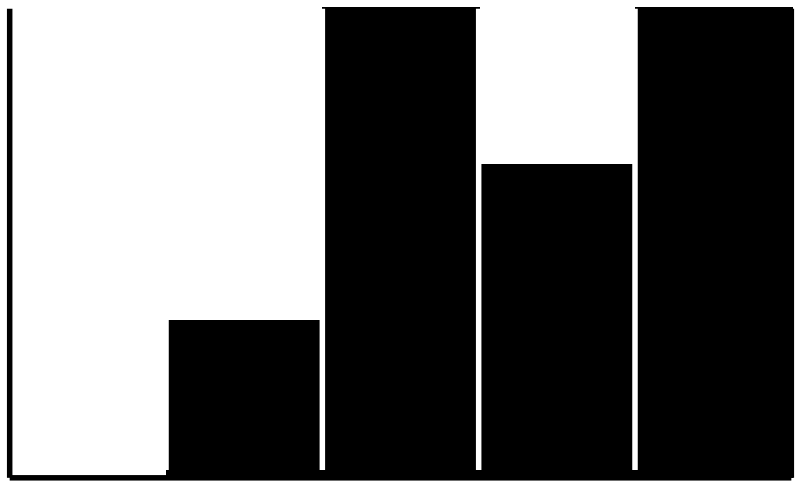
Definition of Color Histograms

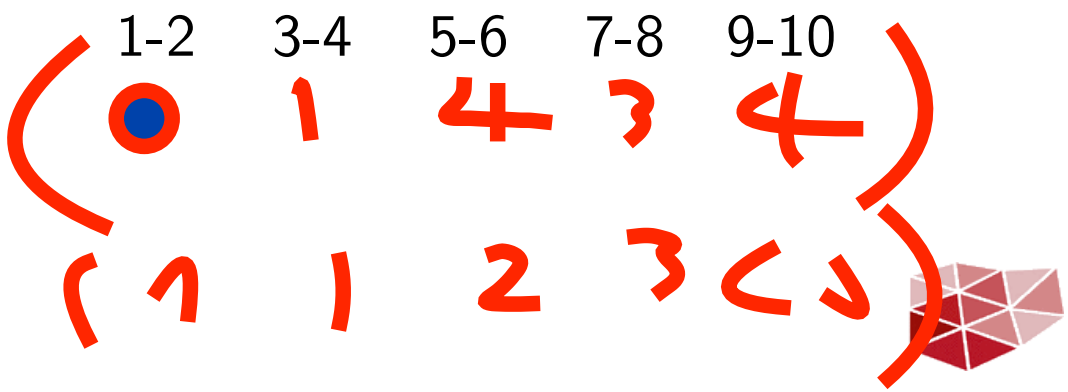


Definition of Color Histograms



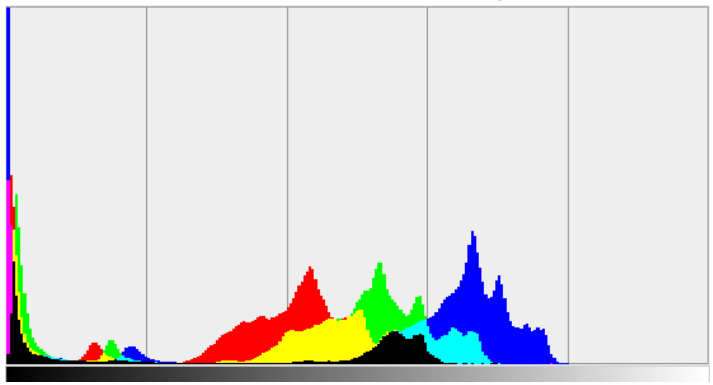
Definition of Color Histograms



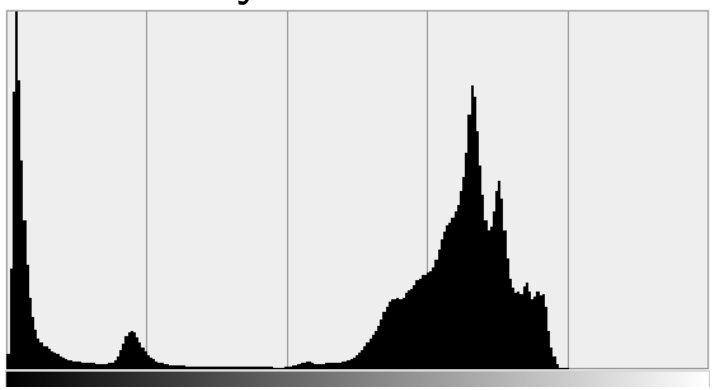


Examples

RGB-Overlay

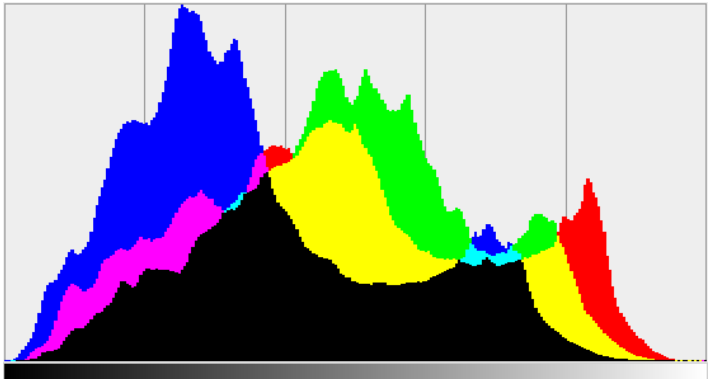


Grayscale-Values

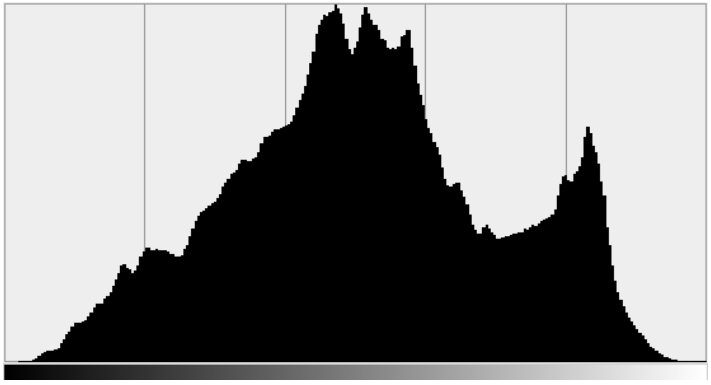


Examples

RGB-Overlay

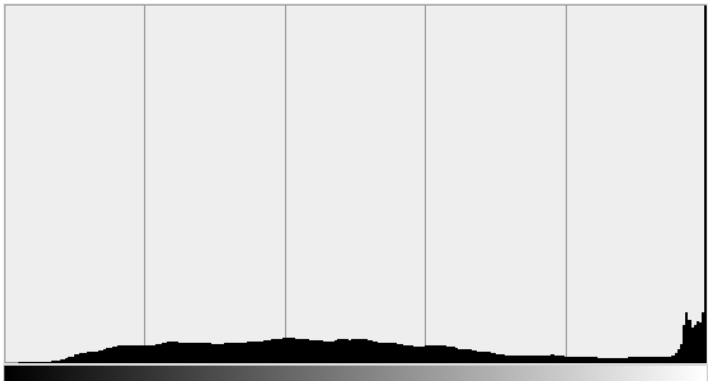


Grayscale-Values

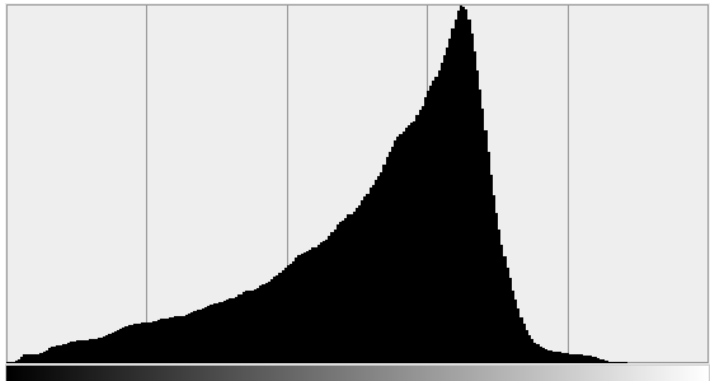


Examples

Grayscale-Values



Grayscale-Values



Advantages and Disadvantages

Advantages

- Translation and rotation invariant
- Colors often represent content

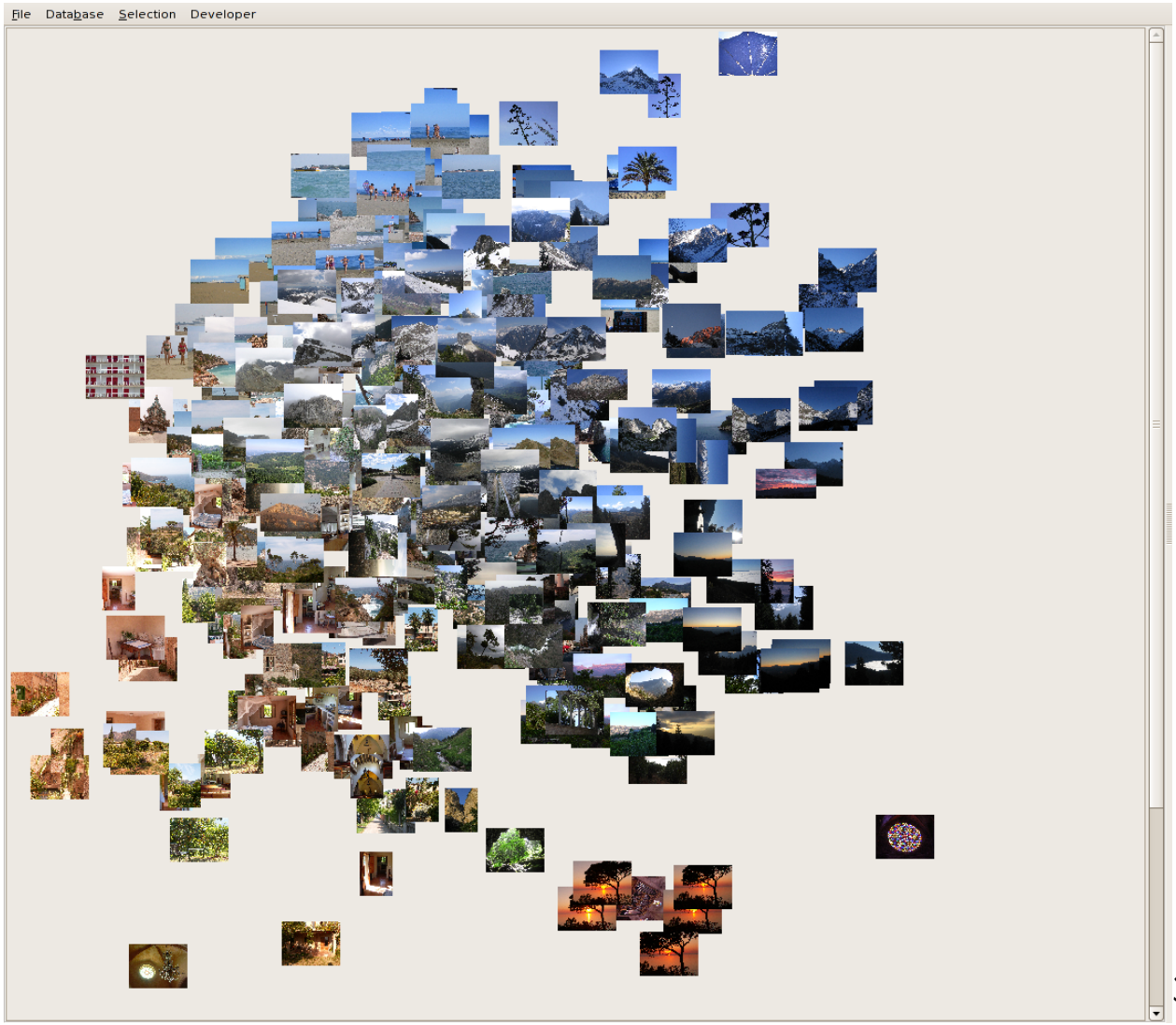
Disadvantages

- Translation and rotation invariant
- No consideration of features and structures

Distance between Color Histograms

- Simple approaches (SSD,. . .)
- Multi-Dimensional-Scaling

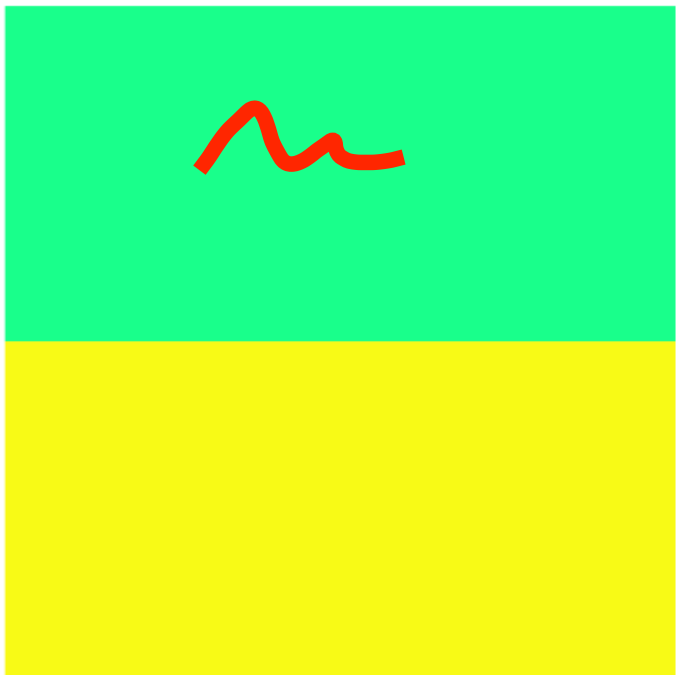
Use of Color Histograms

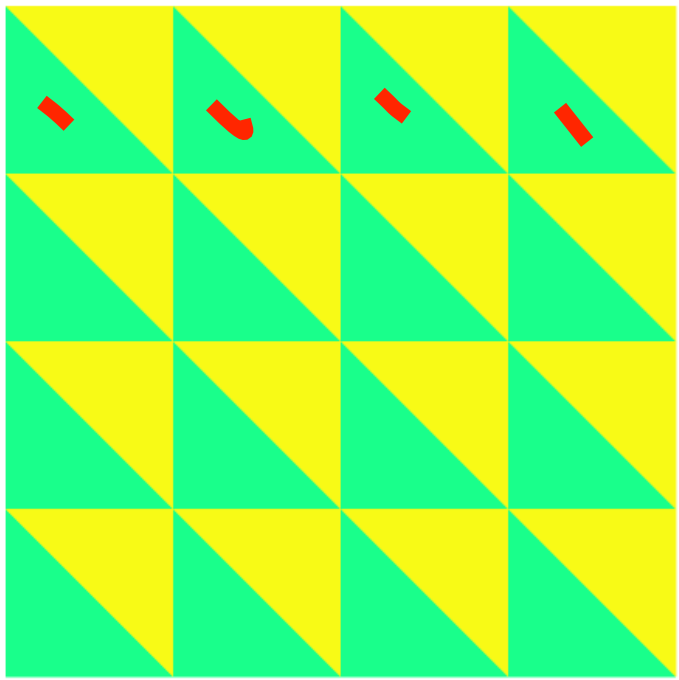


[3]

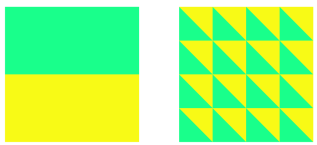
[3] http://lear.inrialpes.fr/src/yorg/

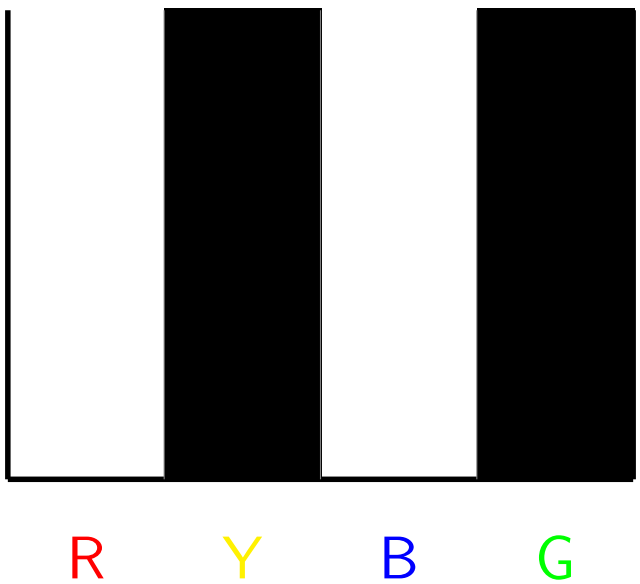
The Problems of Color Histograms



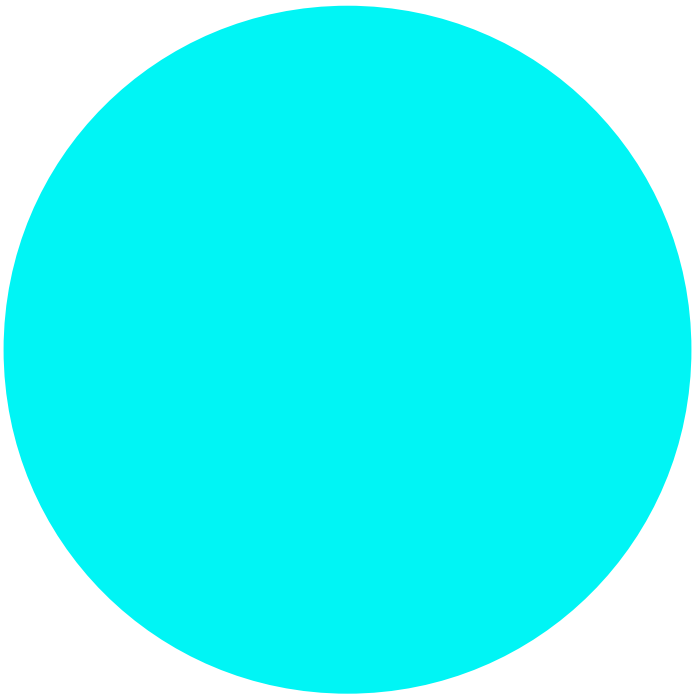


The Problems of Color Histograms





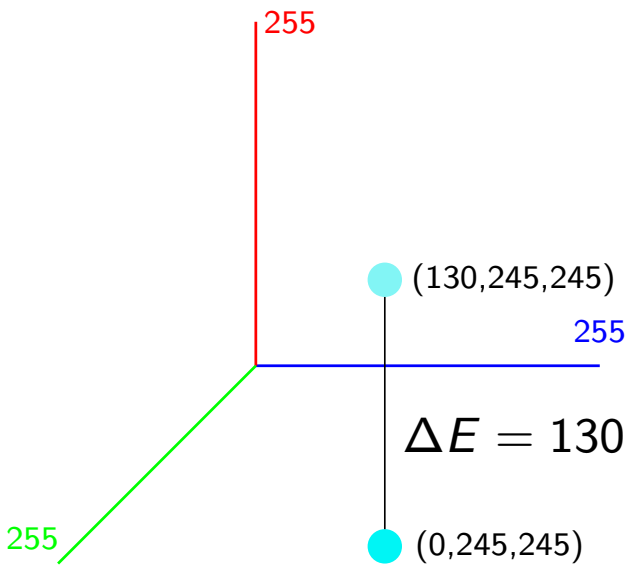
The Importance of Color Spaces



The Importance of Color Spaces

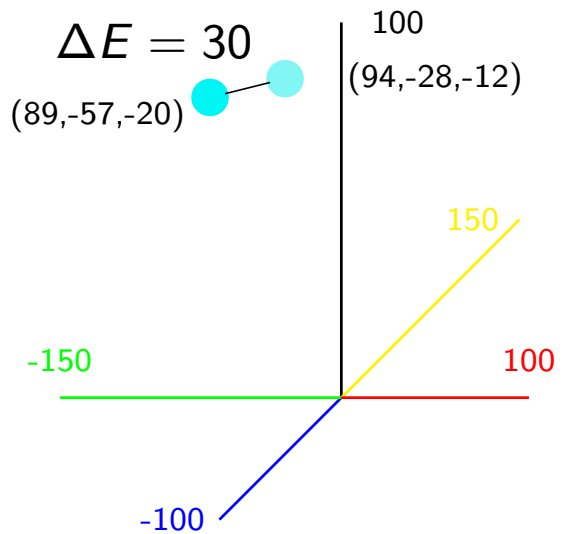
RGB

- Models the output on phys. devices
- Euclidean distances are pointless



Lab

- Designed to approximate human vision
- Euclidean d. represents human perception



$$\Delta E = \sqrt{\Delta L^2 + \Delta a^2 + \Delta b^2}$$

Outline

Gist Descriptor

- gist = quintessence; general idea
- created for recognition of similar scenes, like mountains, tall buildings, streets ... hence the name
- structure, without color information

Gist Descriptor

- gist = quintessence; general idea
- created for recognition of similar scenes, like mountains, tall buildings, streets ... hence the name
- structure, without color information

Gist Descriptor

- gist = quintessence; general idea
- created for recognition of similar scenes, like mountains, tall buildings, streets ... hence the name
- structure, without color information

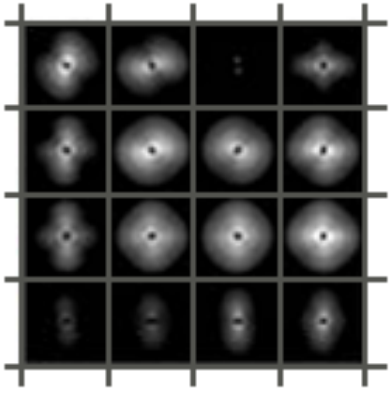
Different Scenes

General Idea

Idea:
Partition the image to anchor the structure of the subimages to their location in the image.

split

fourier transformation

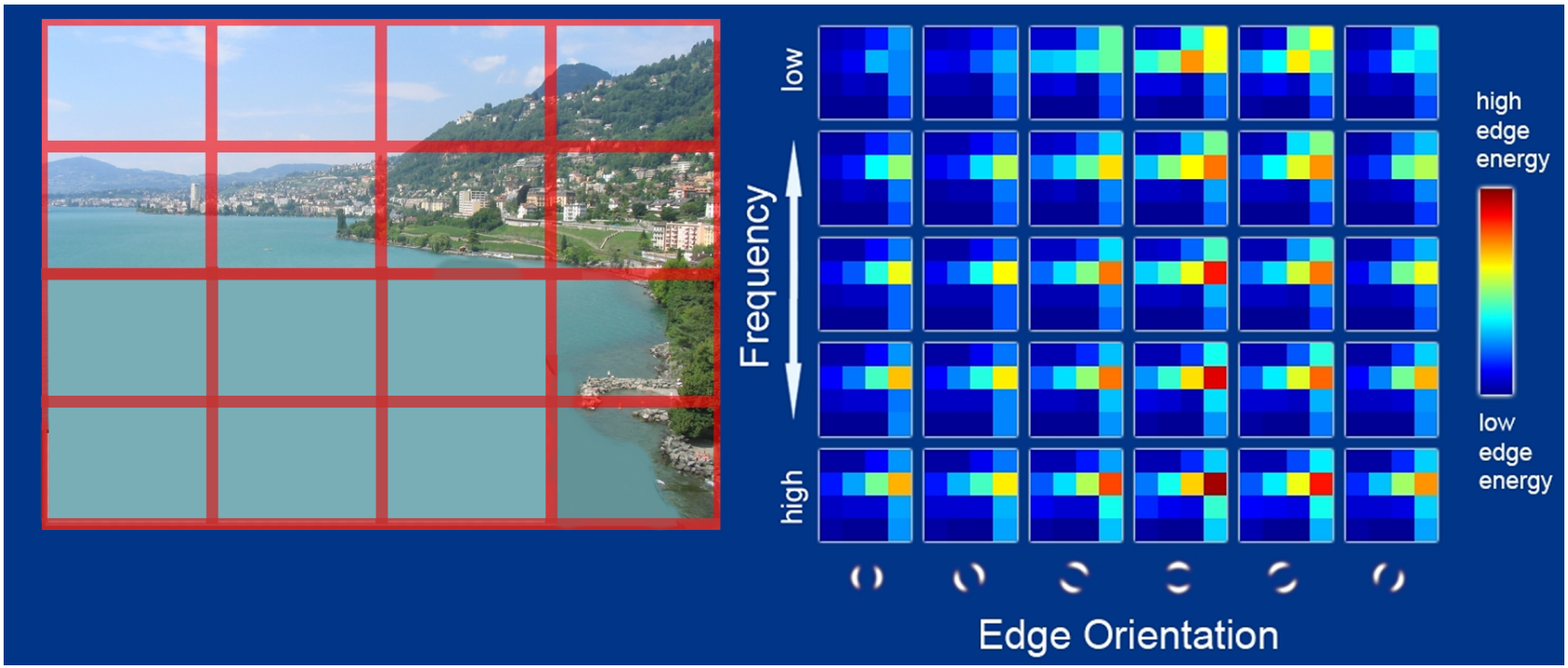


multiscale oriented filters

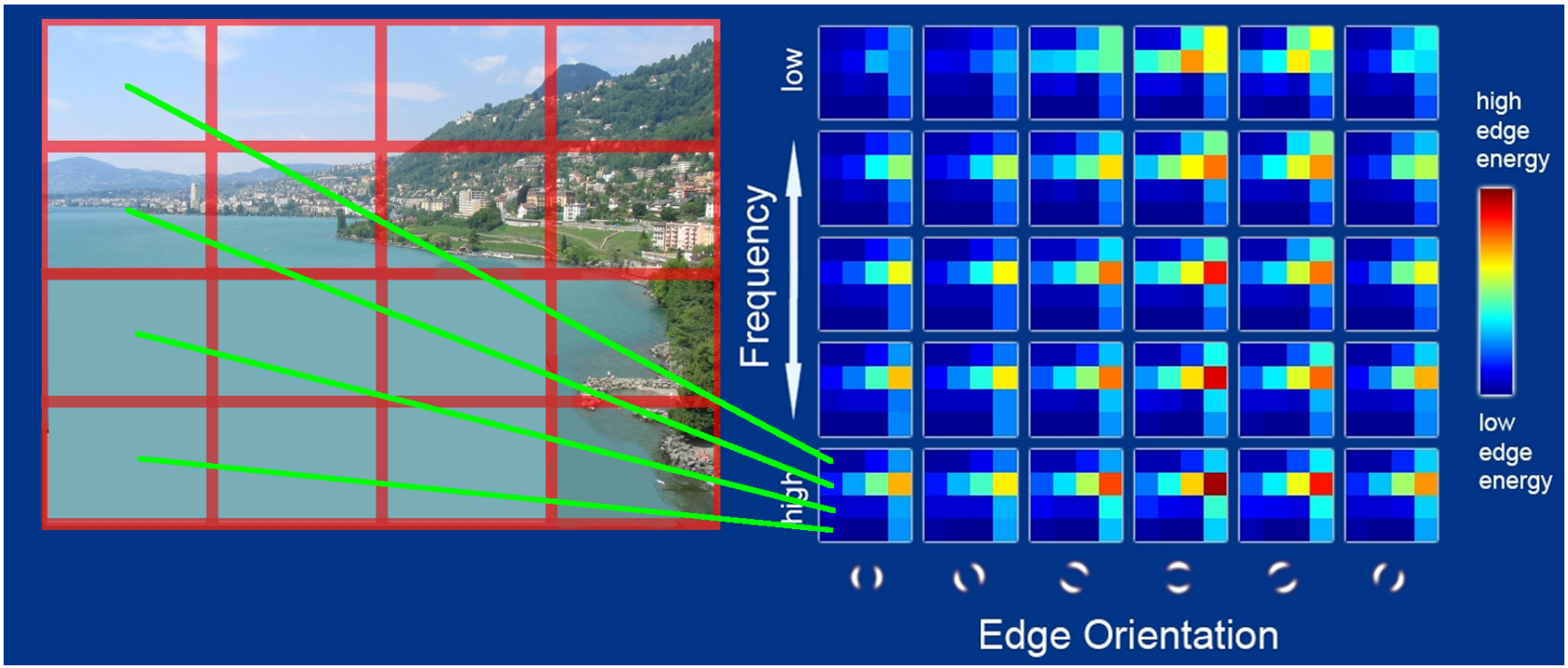
n x n x k - vector
=
gist-descriptor

n x n = # partitions
k = #scales x #orientations

Gist Descriptor

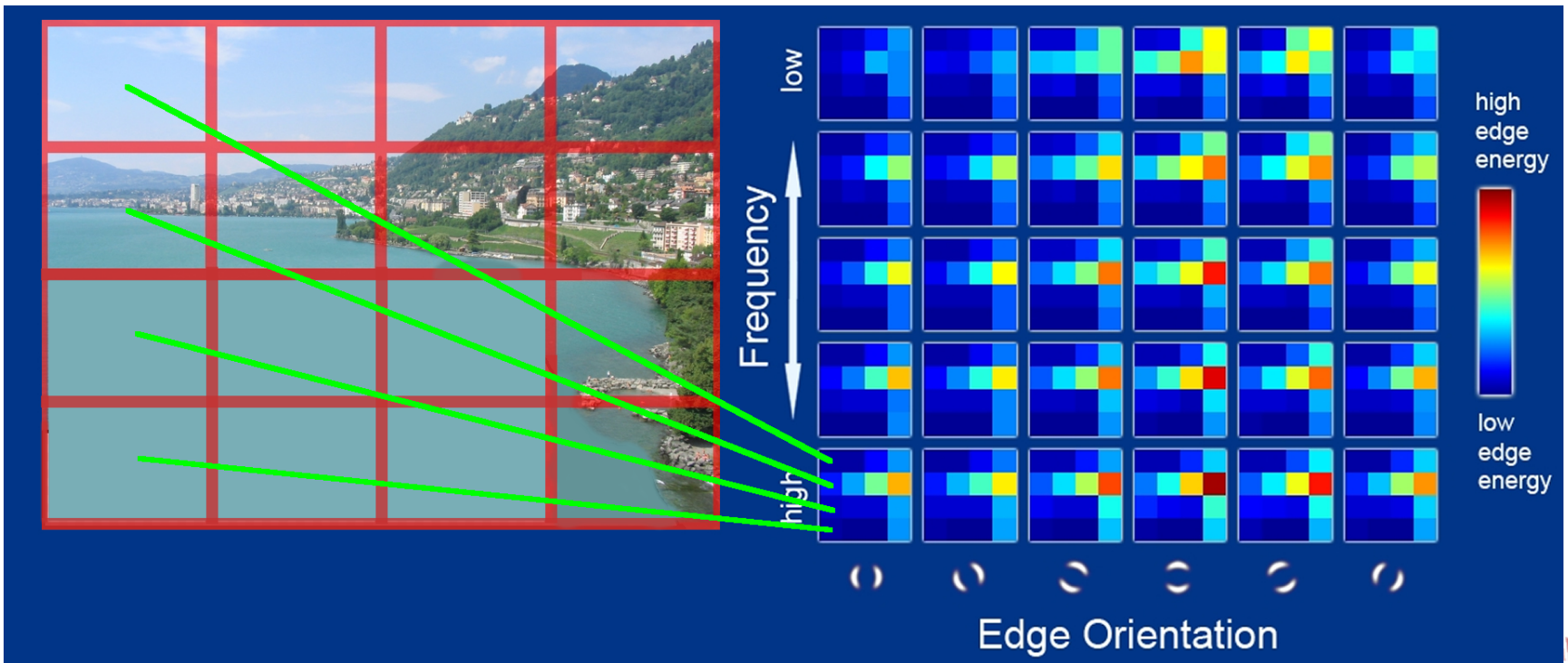


Gist Descriptor

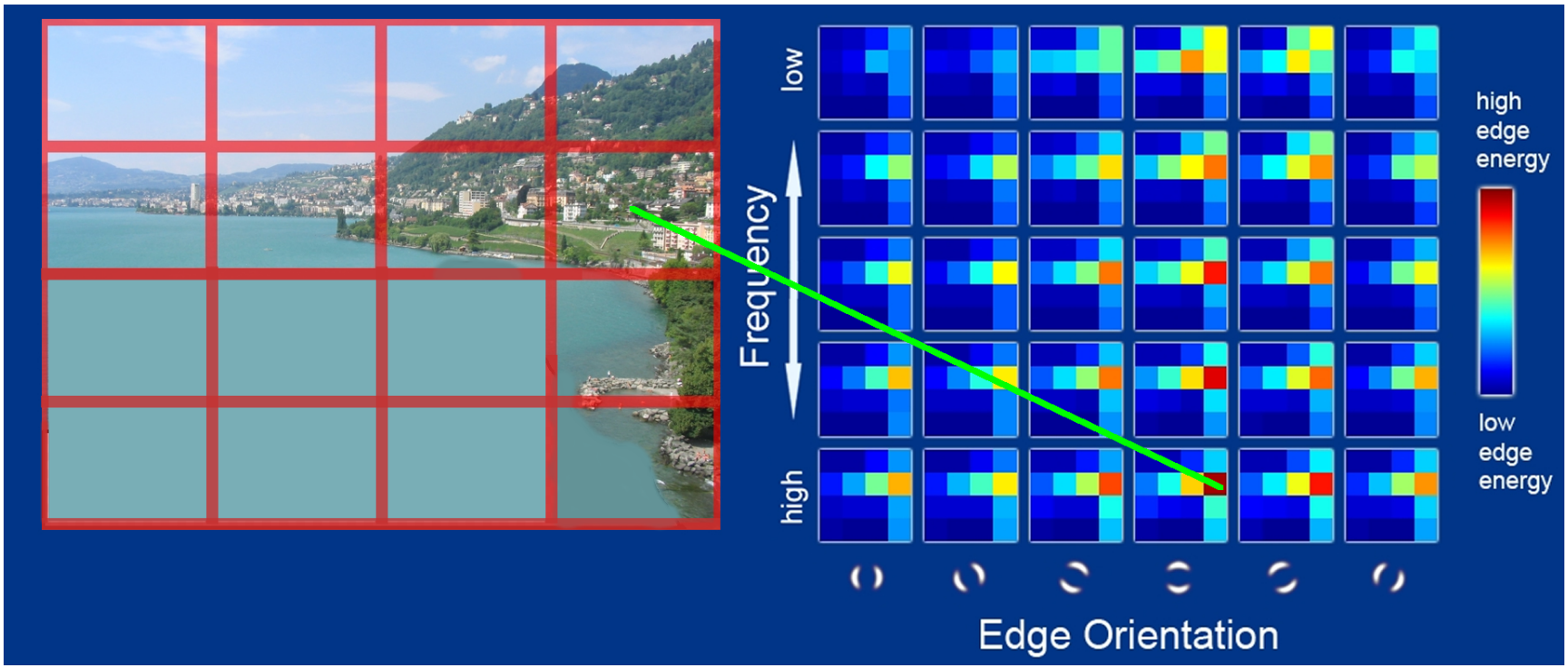


Gist Descriptor

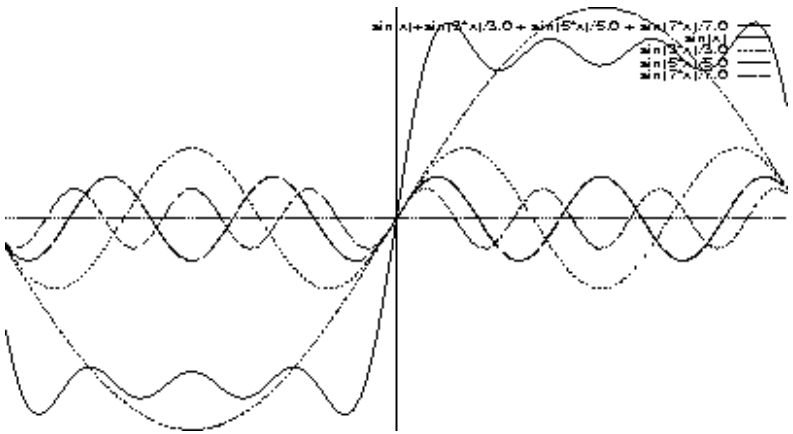
n x n x k = 4 x 4 x 5 frequencies x 6 orientations
= 480-dim. Vector



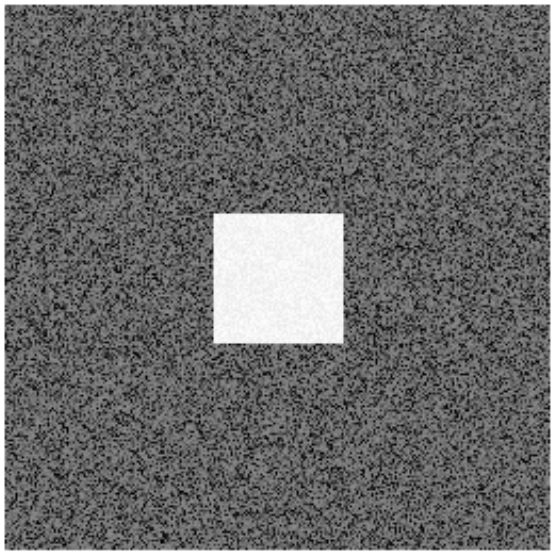
Gist Descriptor



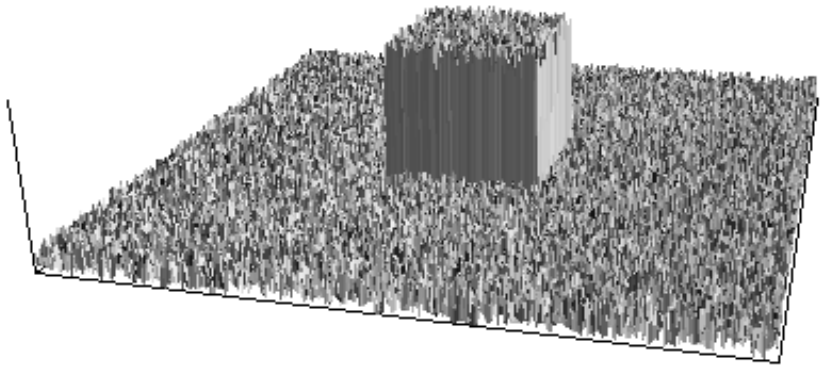
Fourier Transformation



rectangular pulse (with added noise)

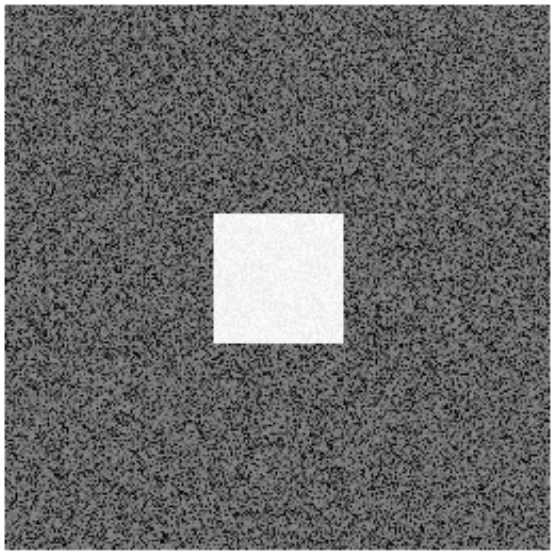


image

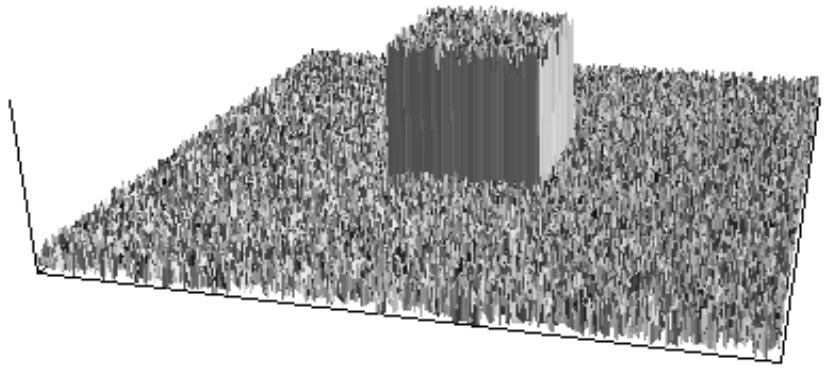


isometric view

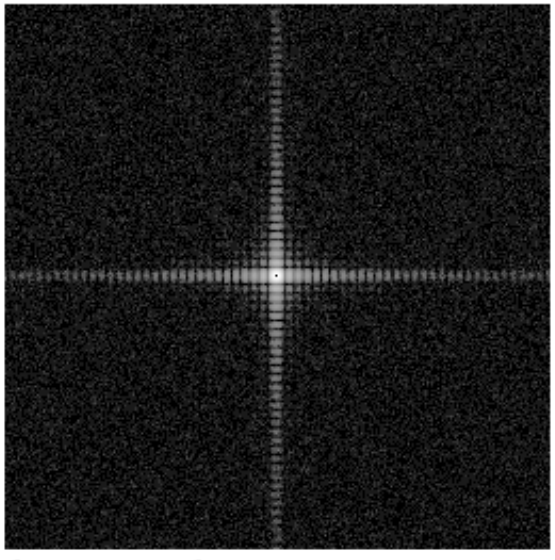
rectangular pulse (with added noise)



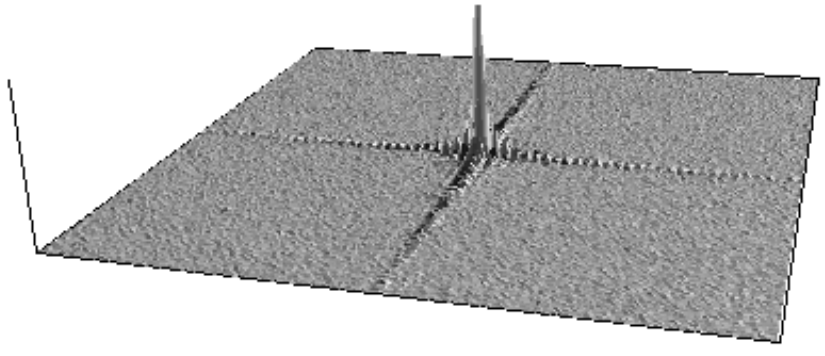
image



isometric view

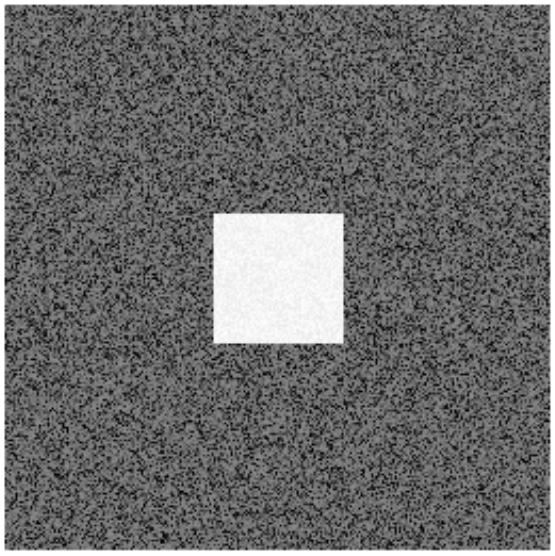


fourier transformation

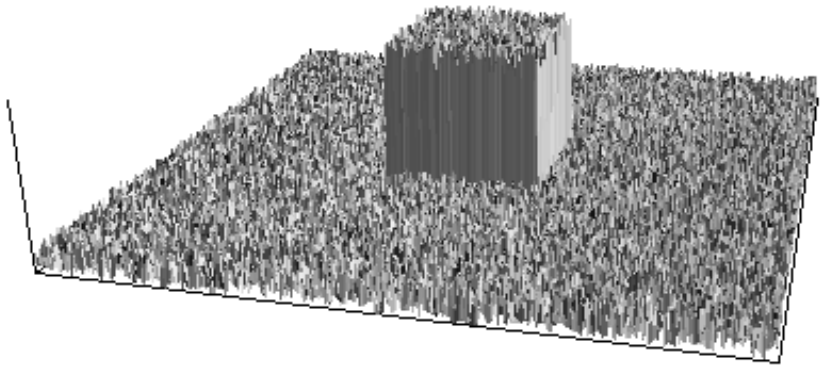


isometric view

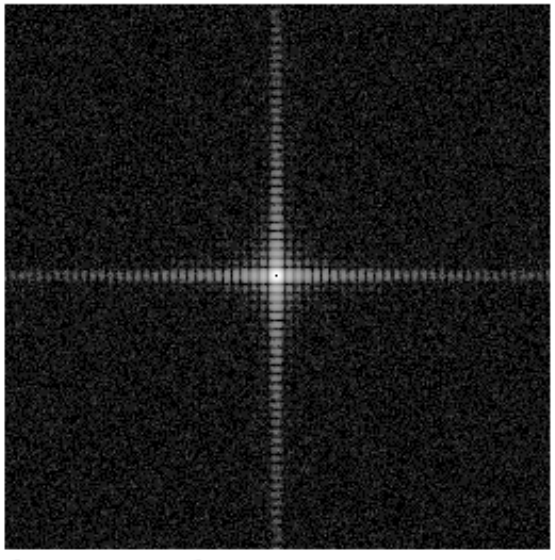
rectangular pulse (with added noise)



image

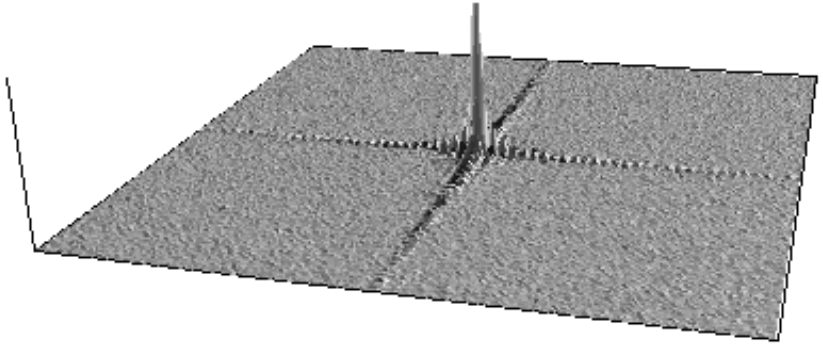


isometric view



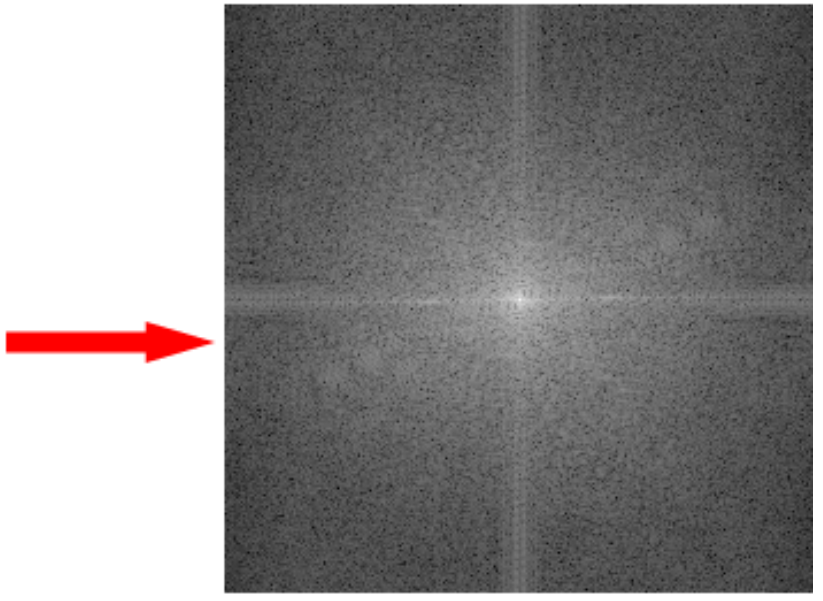
fourier transformation

only frequency and amplitude!



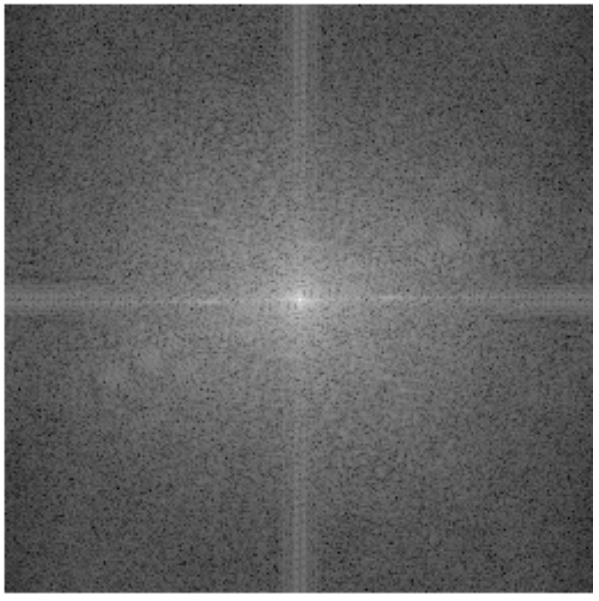
isometric view

image

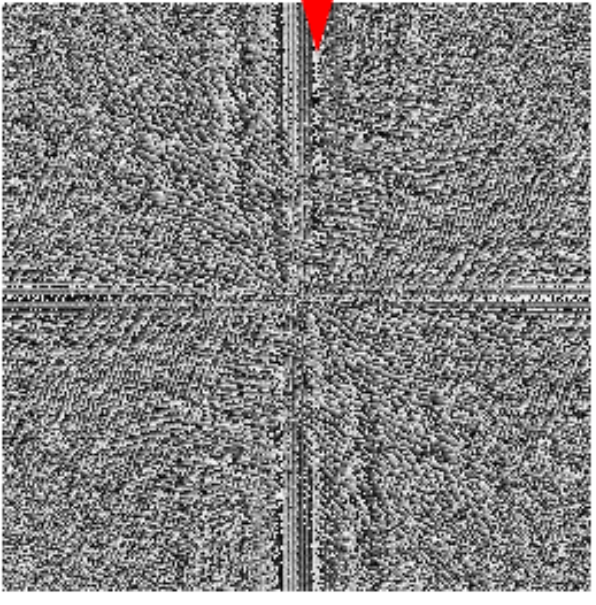


fourier transformation (amplitude over frequency)

image

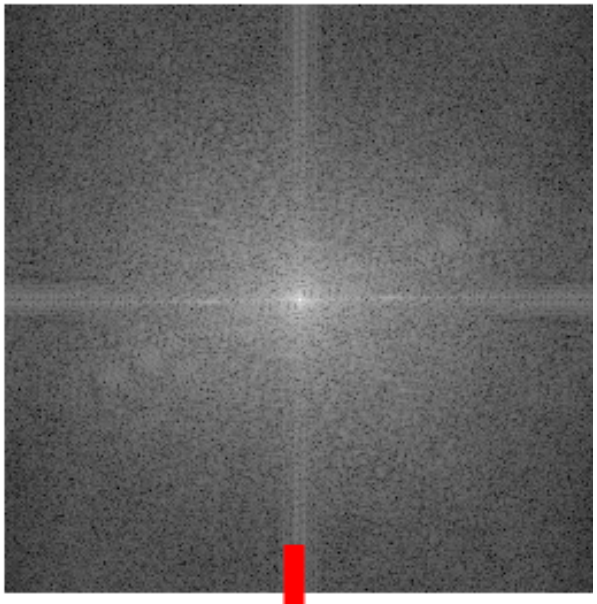


fourier transformation (amplitude over frequency)

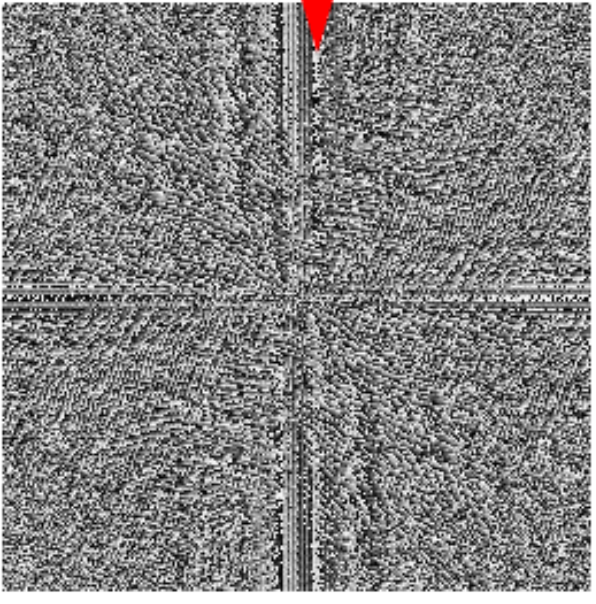


fourier transformation (phase over frequency)

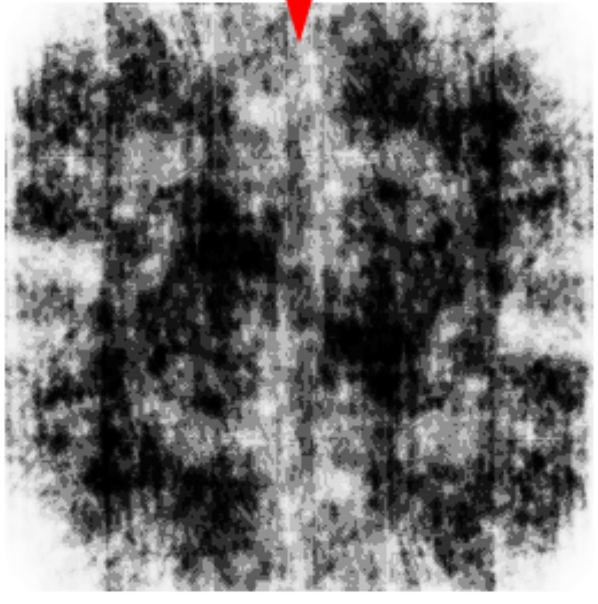
image



fourier transformation (amplitude over frequency)



fourier transformation (phase over frequency)



retransformation with phases = 0

fourier transformation

fourier transformation

fourier transformation

amplitudes

fourier transformation

phases

retransformation

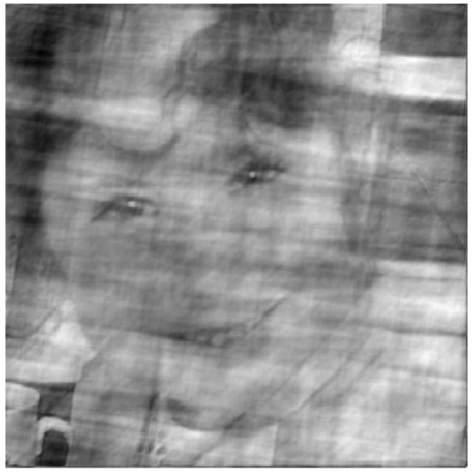
fourier transformation

fourier transformation

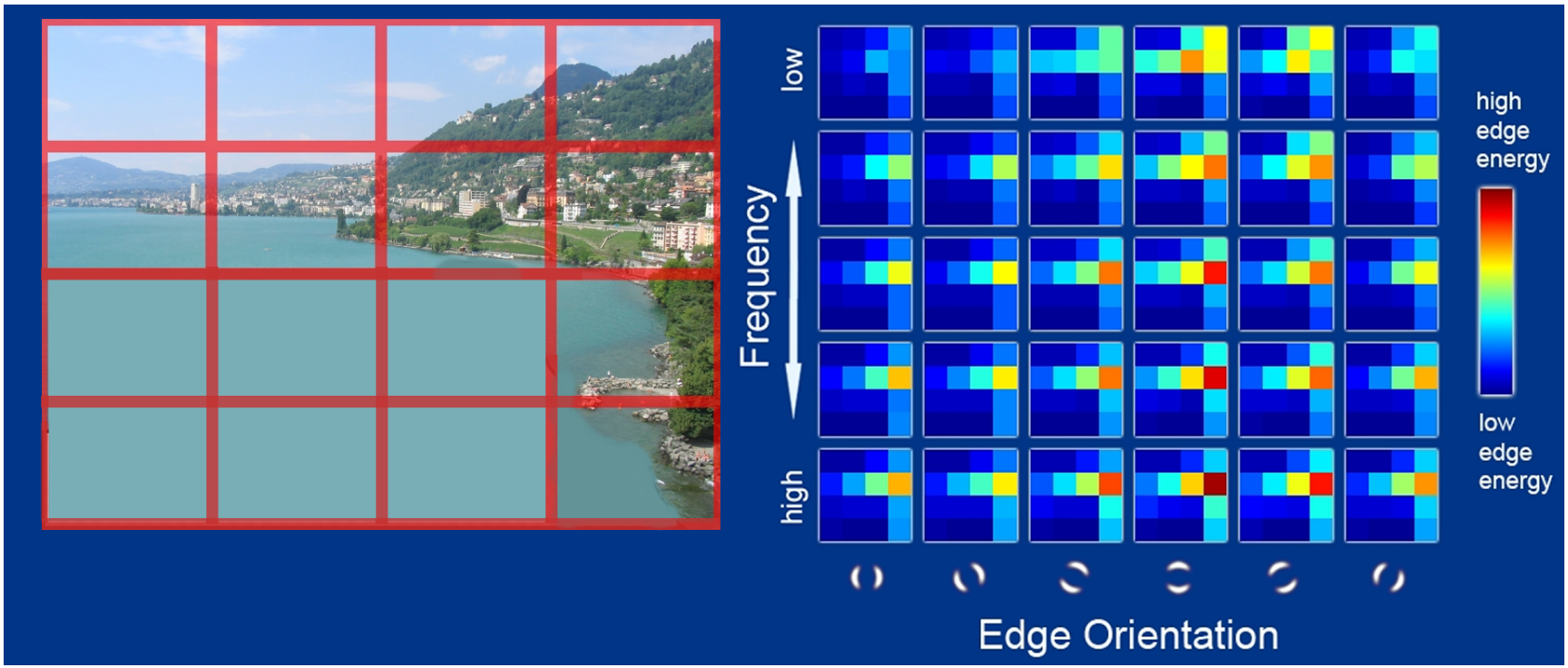
amplitudes

phases

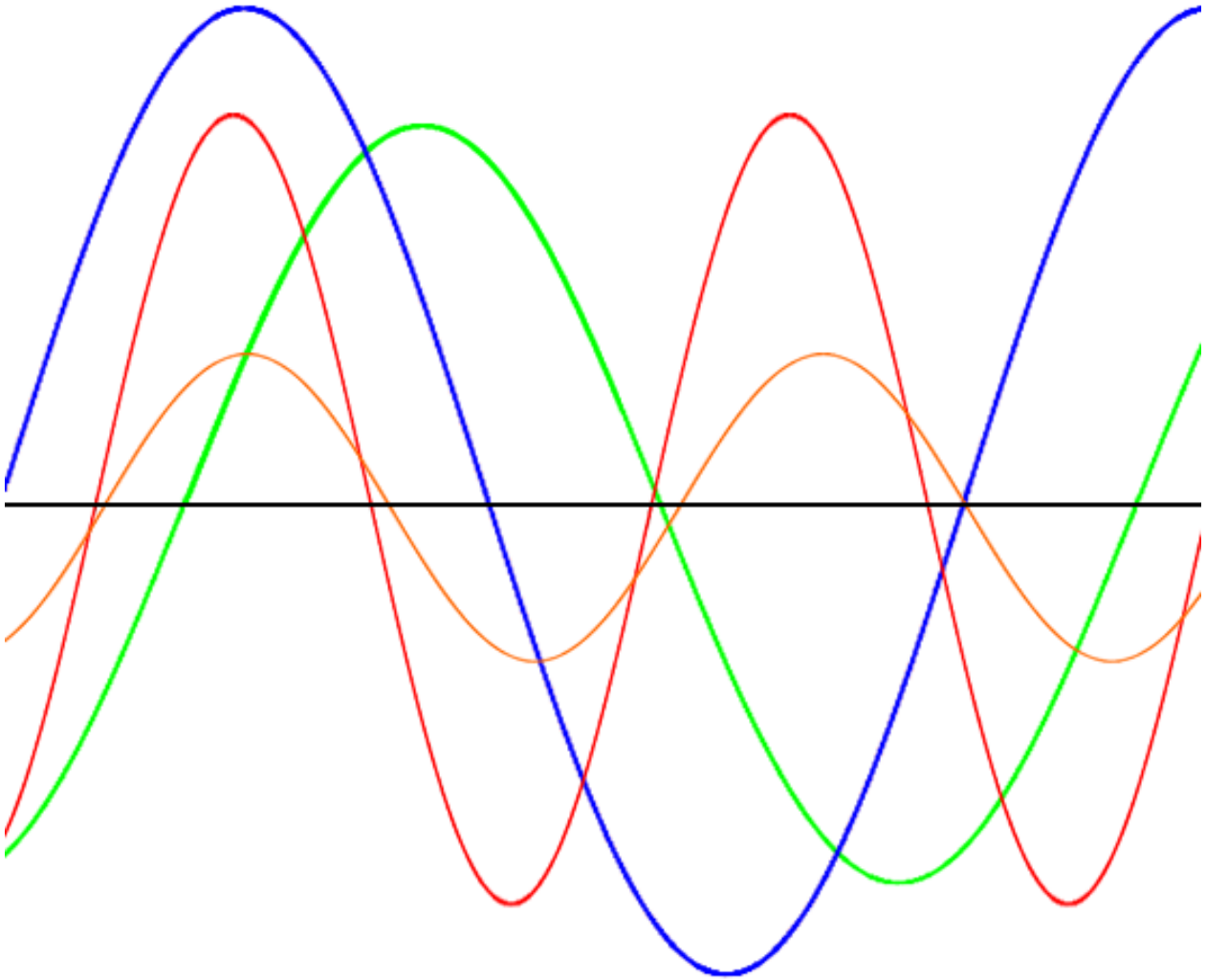
retransformation



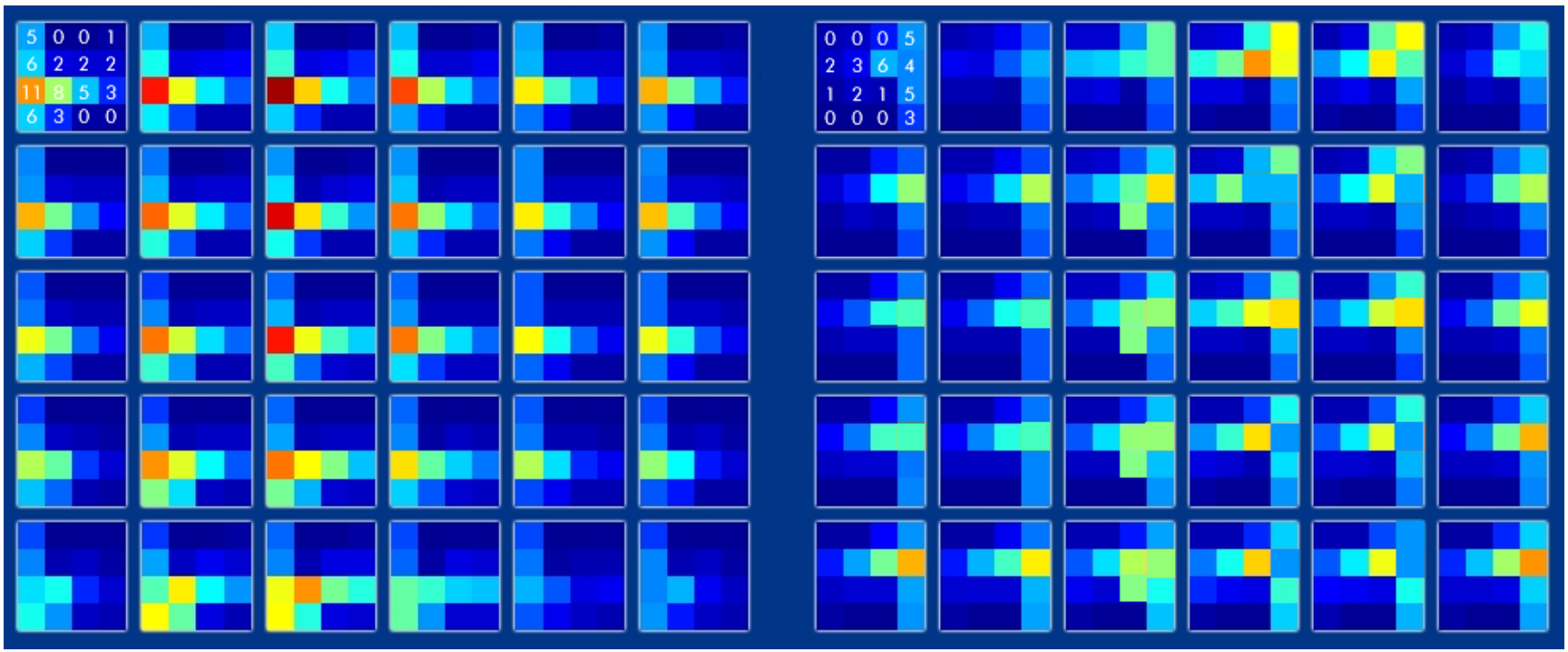
Gist Descriptor



How to cluster



SSD to compare gist descriptors



Advantages and Disadvantages

Advantages

- holistic approach, without segmentation or grouping operations
- get the gist of a scene in a small vector
- easy to store
- easy to handle with established algorithms
- fast

Disadvantages

- difficult implementation

Outline

In Conclusion

- descriptors are very important for image retrieval
- there are different descriptors for different purposes
- descriptors are thus dependent on the problem

Comparison

Descriptor	Advantages	Disadvantages
Tiny images	Easy to calculate	No features
Histograms	Translation/rotation invariant	No local features
Gist	Preserves features and locales	Implementation