BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Learning from Images

Recap and Outlook
Content-Based / Sketch-based
Image / Shape Retrieval

Master DataScience
Winter term 2019/20

Prof. Dr. Kristian Hildebrand
khildebrand@beuth-hochschule.de

Understanding Images / Image Retrieval

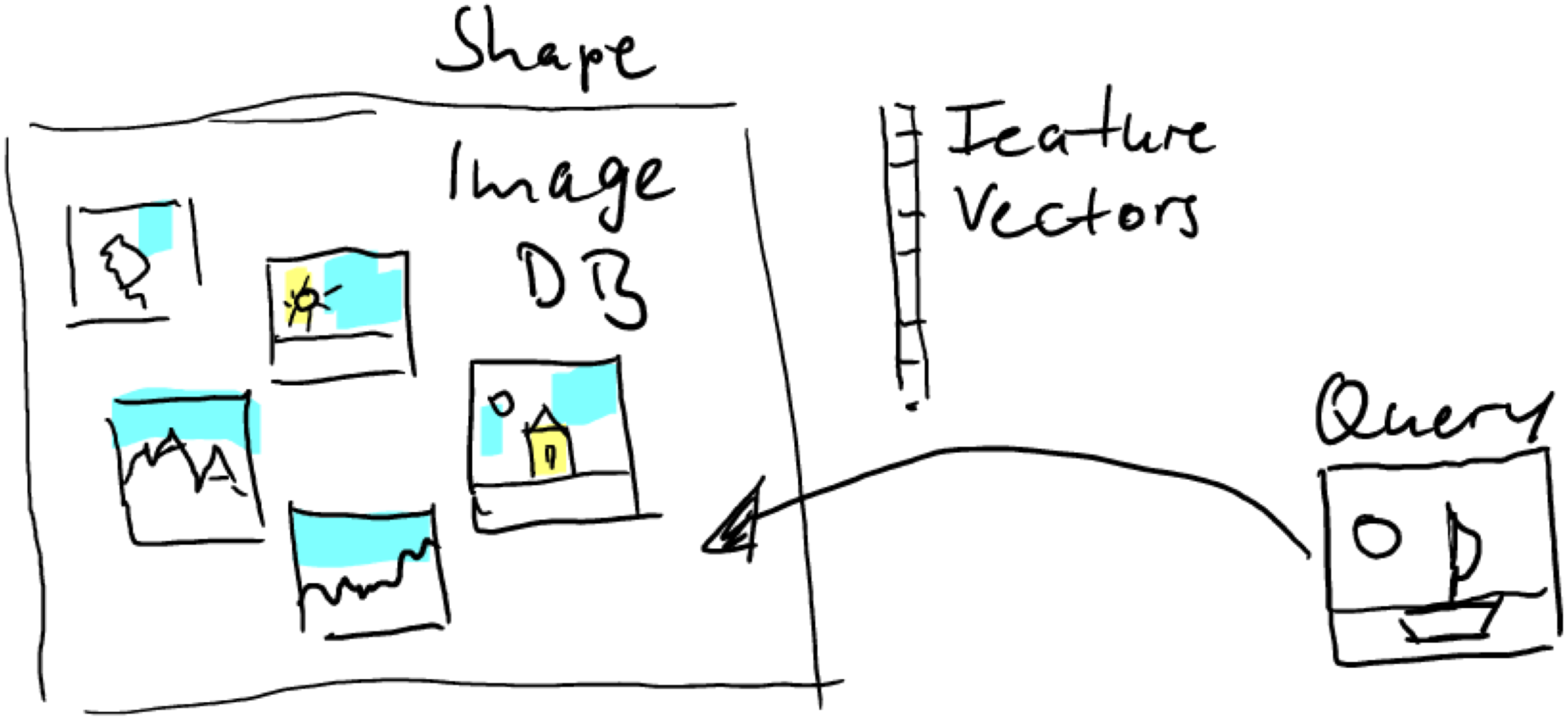
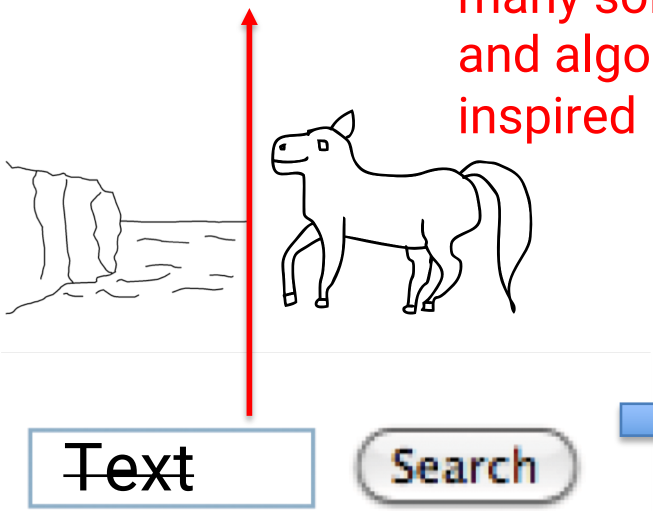
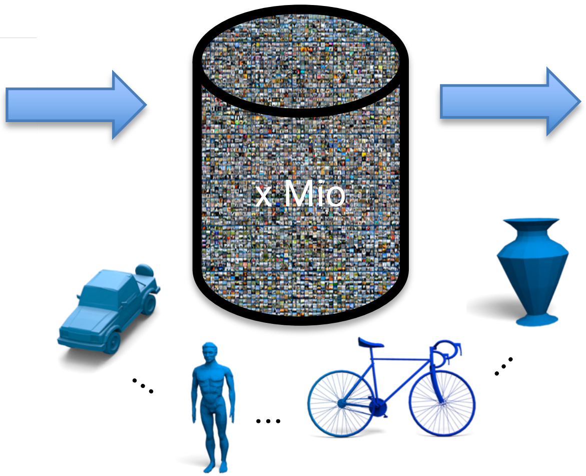


Image Retrieval

Overview

strong connection to text retrieval /
many solutions, data structure and algorithms
inspired by text retrieval research



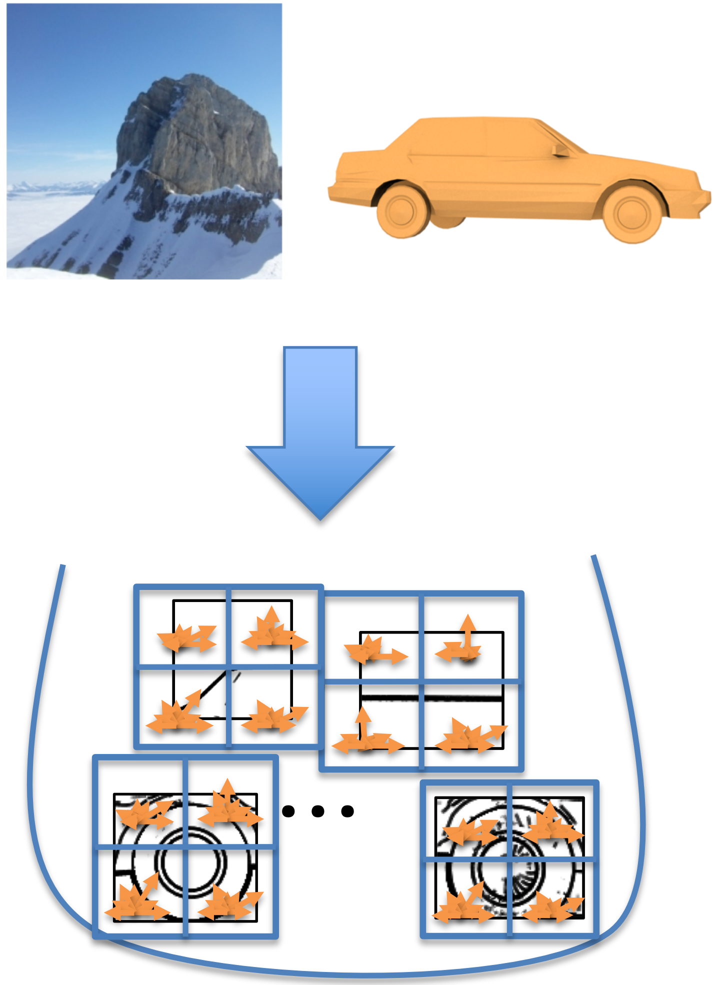


Bag of features

Bag of features

- Orderless document representation: frequencies of words from a dictionary

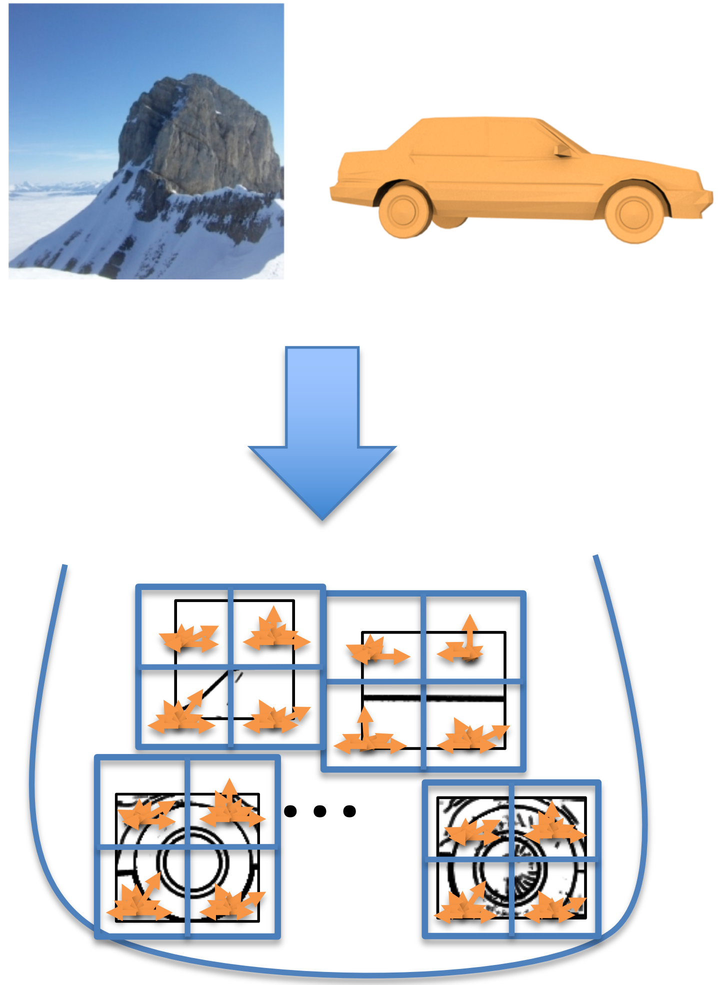
http://3.bp.blogspot.com/-aepyVr_3pt0/U3QB_D_nNVI/AAAAAAABmzU/6xPuRaFH33c/s1600/Tag+cloud.png



Bag-of-features [Sivic'03]

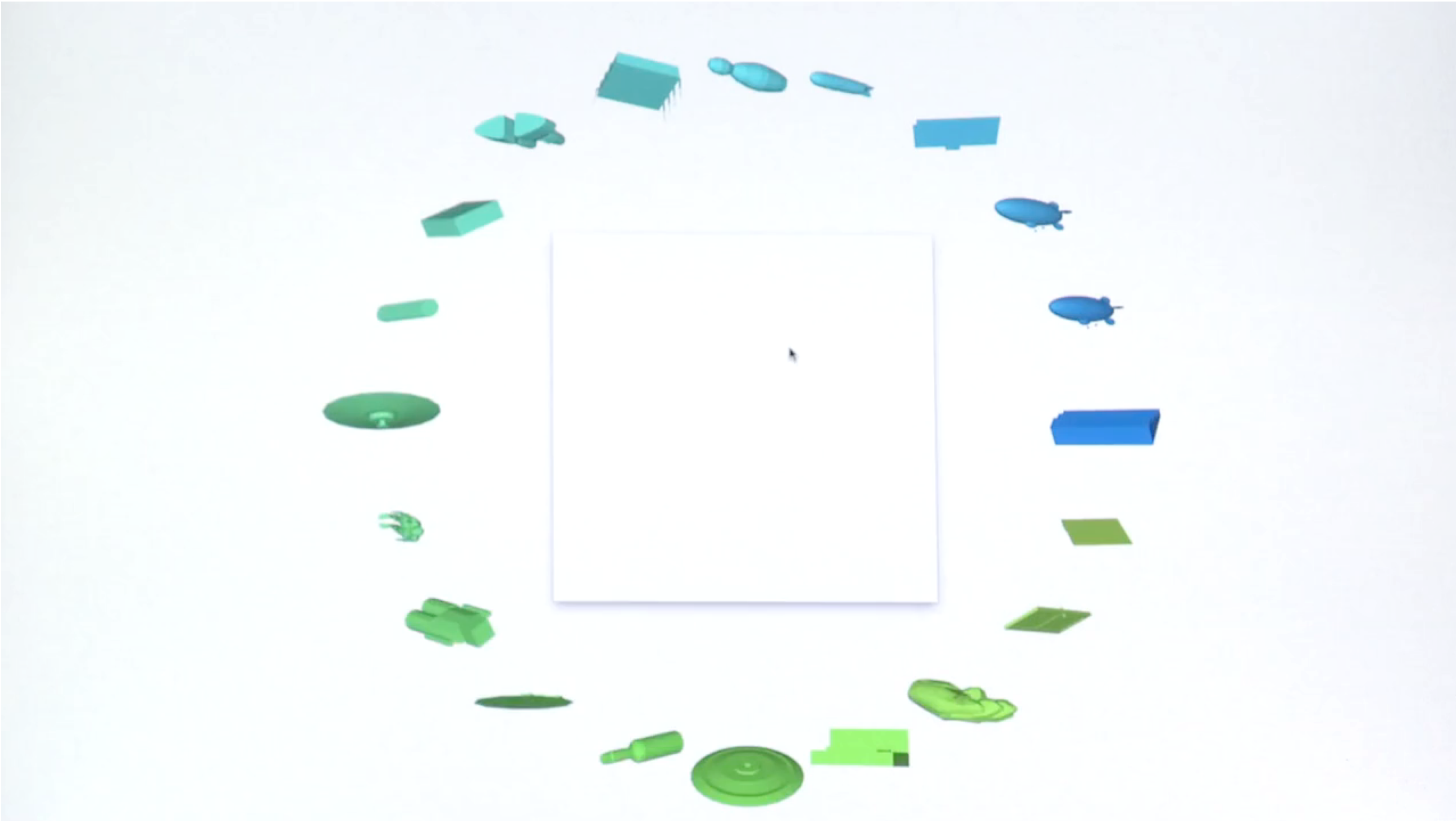
Bag of features

1. **Extract features**
2. Learn “**visual vocabulary**”
3. **Quantize features** using visual vocabulary
4. Represent images by frequencies of “**visual words**”



Sketch-based Shape Retrieval

Sketch-based Shape Retrieval. *ACM Transactions on Graphics, Proc. SIGGRAPH 2012.*
Eitz, Mathias, Richter, Ronald, Boubekeur, Tamy, Hildebrand, Kristian and Alexa, Marc.

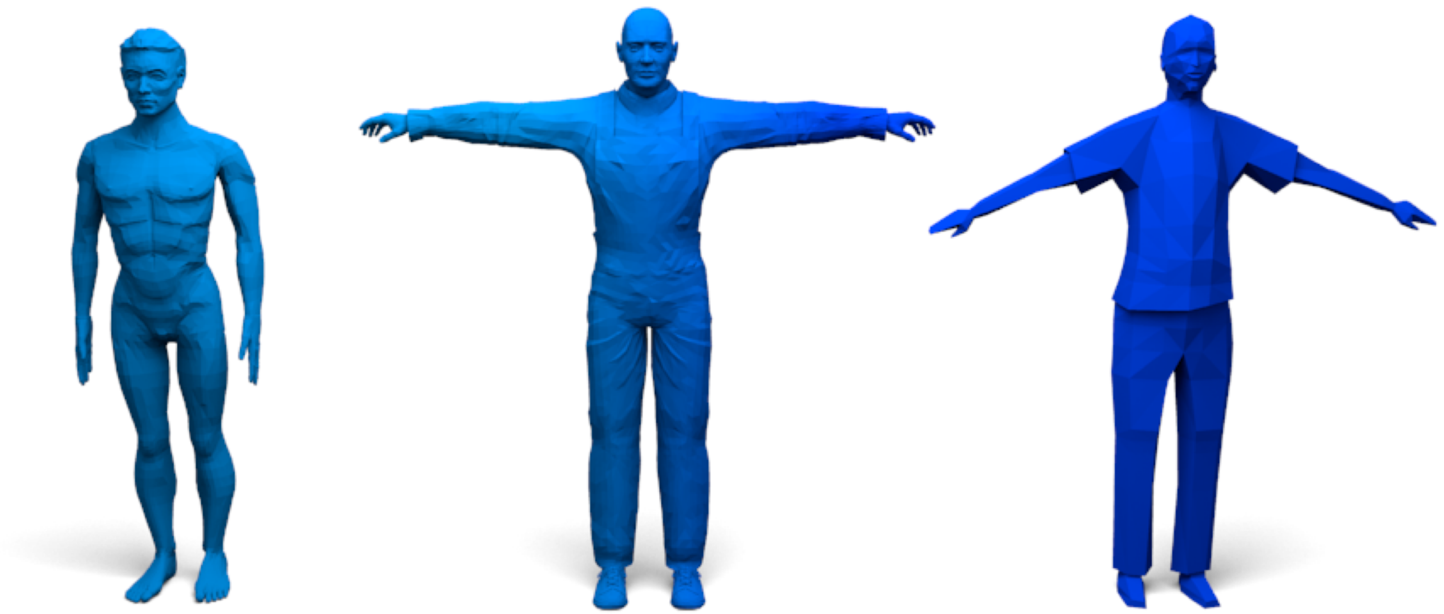


Why Sketch-Based?

- Problems:
  - vehicle, jeep, truck, pickup, ...
  - no keyword attached to model

Why Sketch-Based?

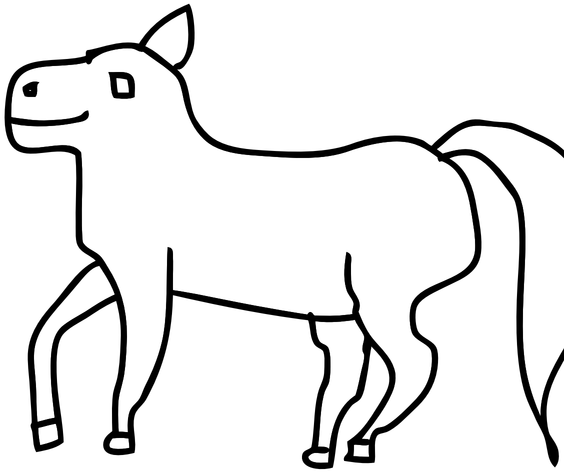
- Easy to sketch, difficult to describe



Why Sketch-Based?

- Easy to sketch, difficult to describe

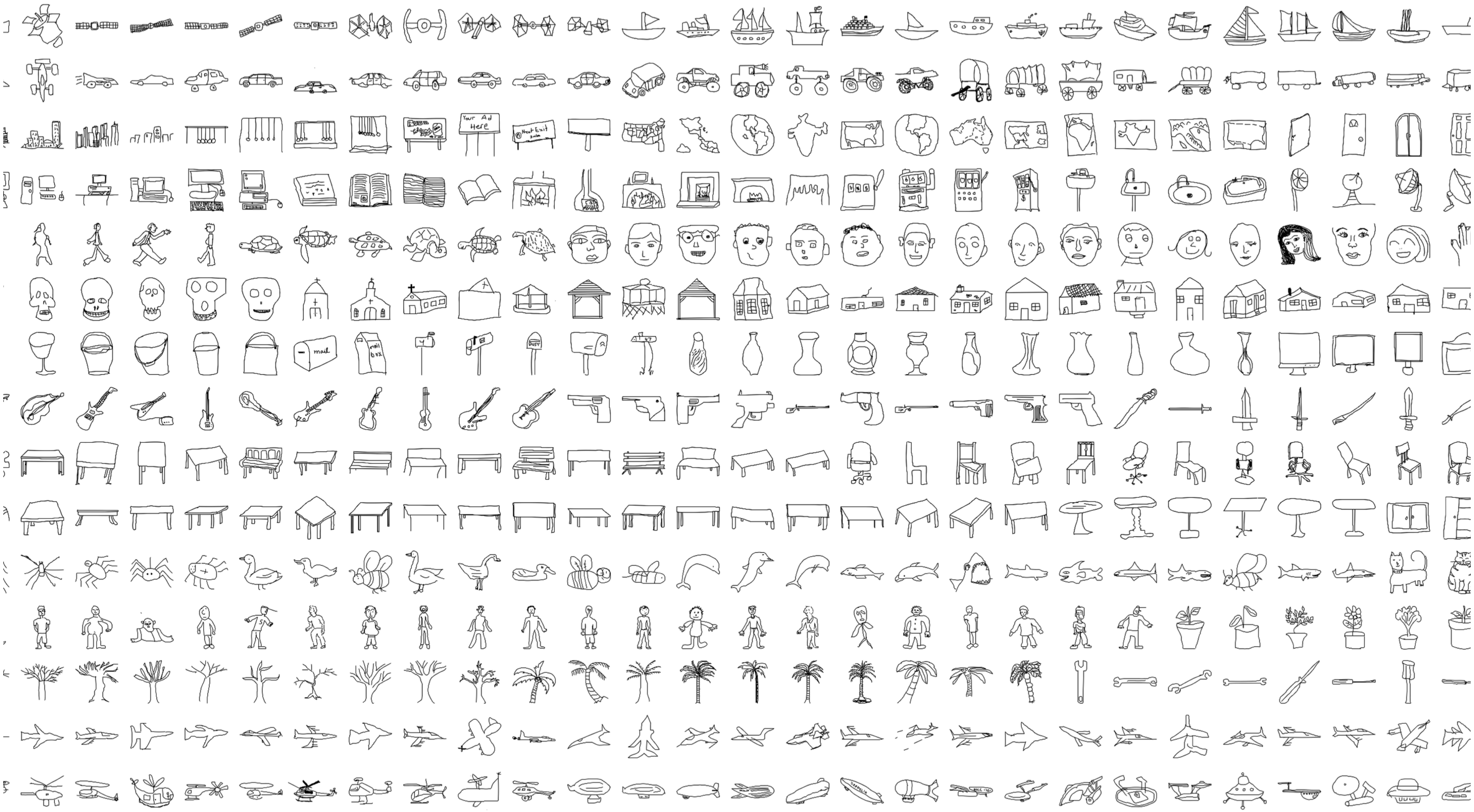
Challenges



- What to match the sketch lines against?
- Sketch is a projection, information lost
- Need to support all possible viewing directions
- Handle extreme abstraction/exaggeration

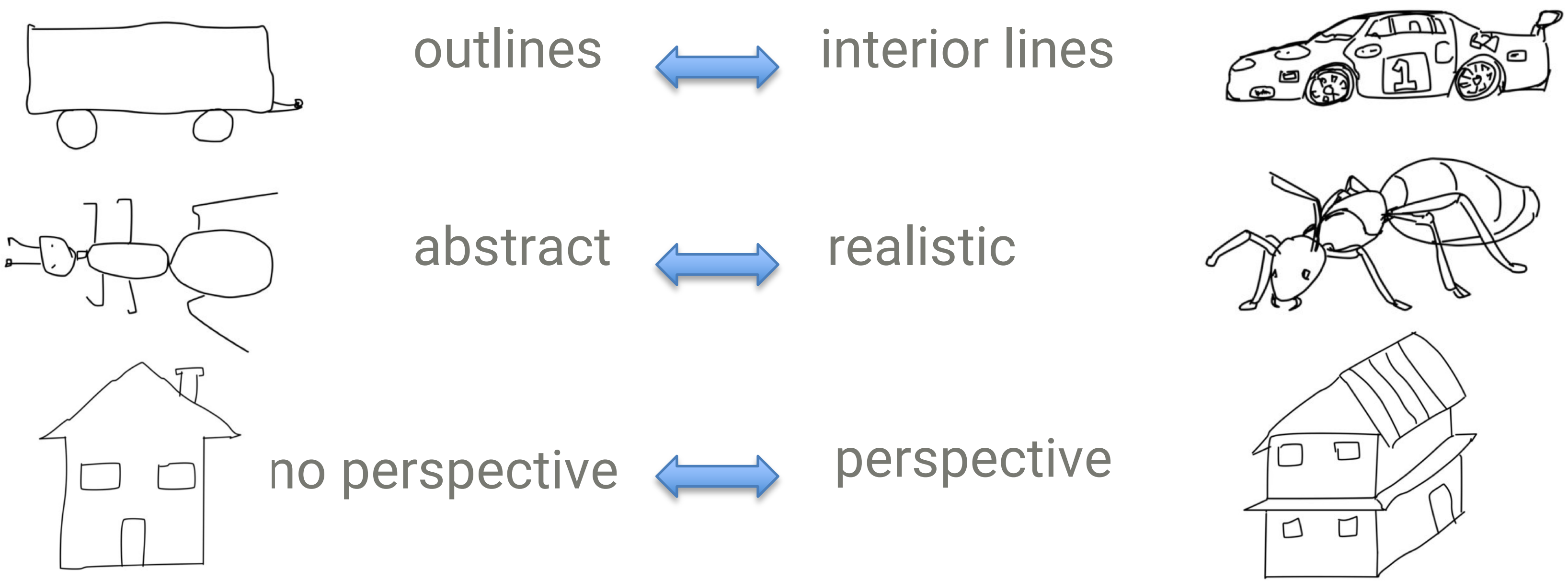
How Do Humans Sketch for Shape Retrieval?

- Questions:
  - Type of lines humans draw, are outlines enough? [Chen 2003]
  - Consistent quality?
  - Realistic/abstract?
- User study on Amazon Mechanical Turk
  - Interactive drawing tool
  - Asked for a total of ~2,000 sketches in 90 categories
  - Categories from Princeton Shape Benchmark [Shilane 2003]


Your Ad Here
mail

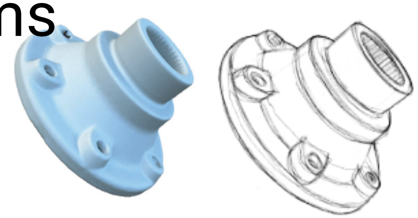
How Do Humans Sketch for Shape Retrieval?

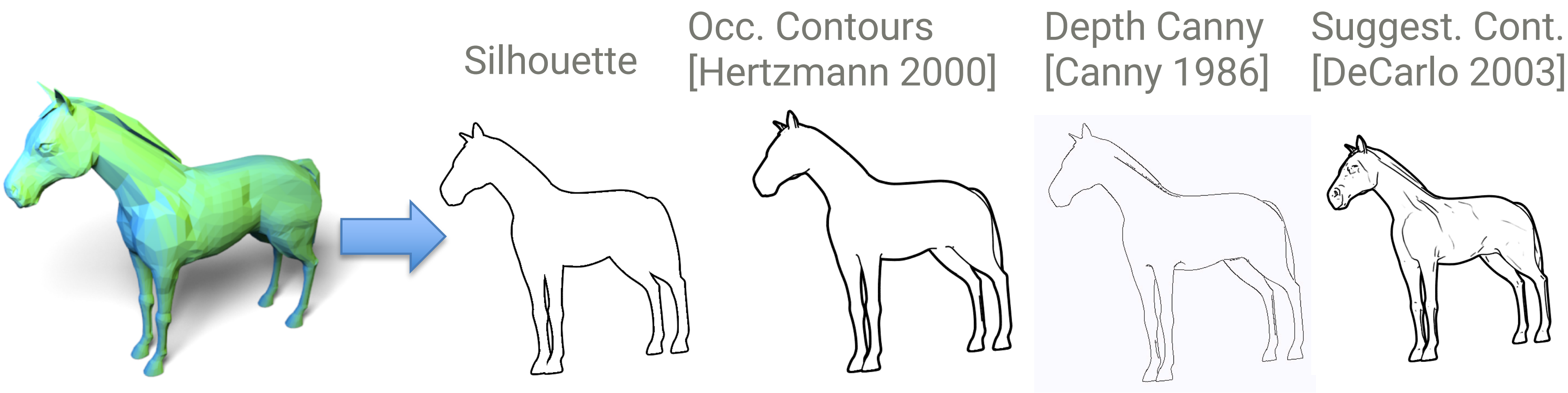
- Large variety of sketching styles:



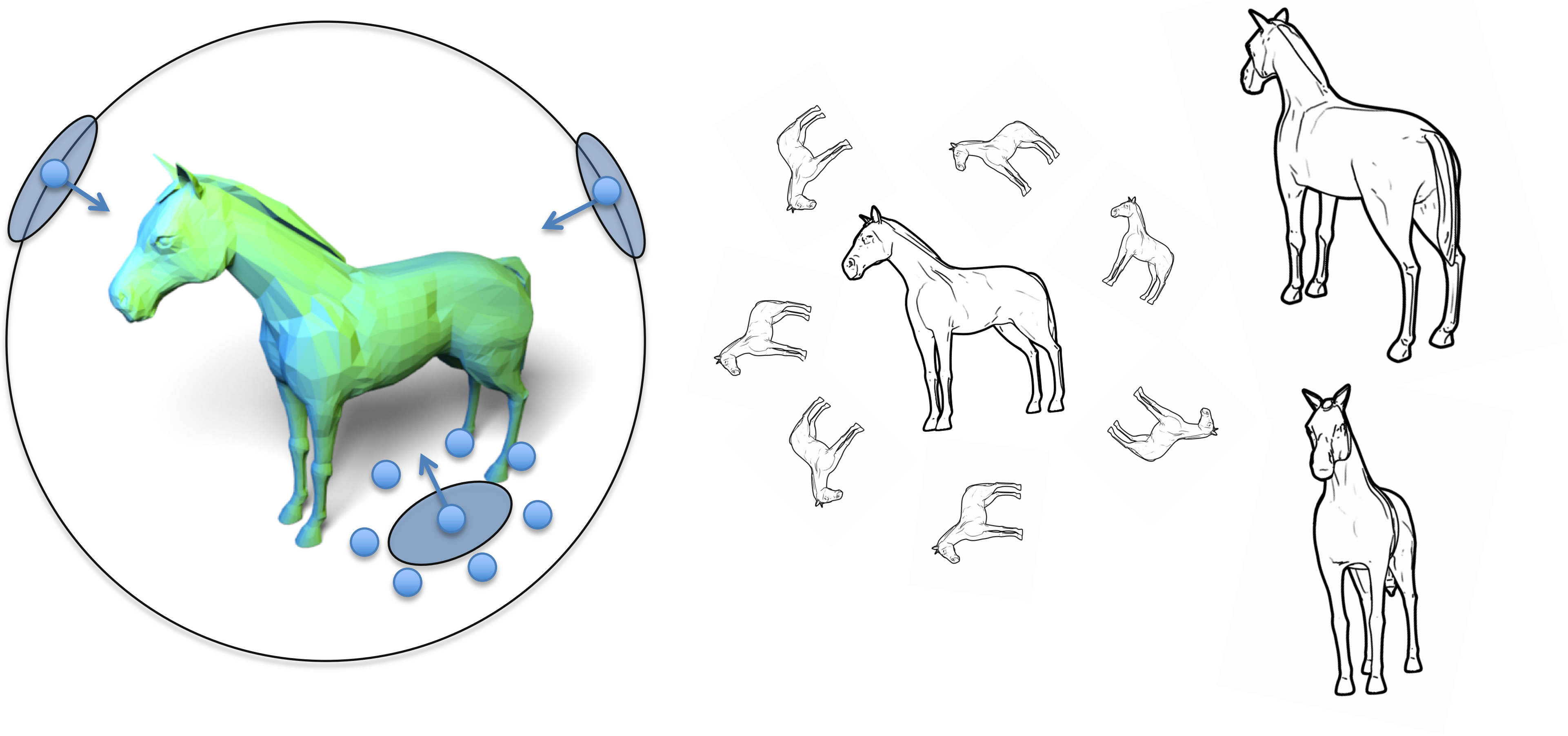
View-Based Approach

- View-based instead of direct matching to 3D shape
  - [Bülthoff‘92]: humans represent shapes using 2D views
  - [Cole‘07]: 90% of lines explained by NPR algorithms



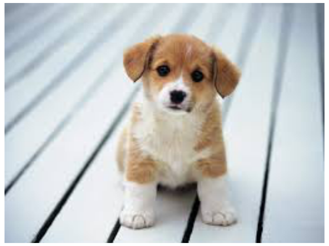


View Generation

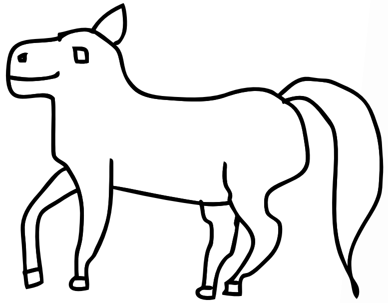


Similarity Measure

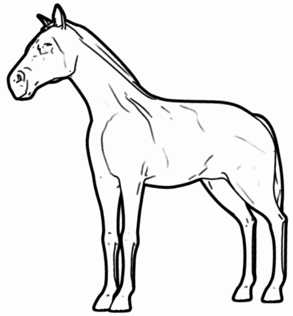
Image-based retrieval



similarity (



,



)

Requirements:

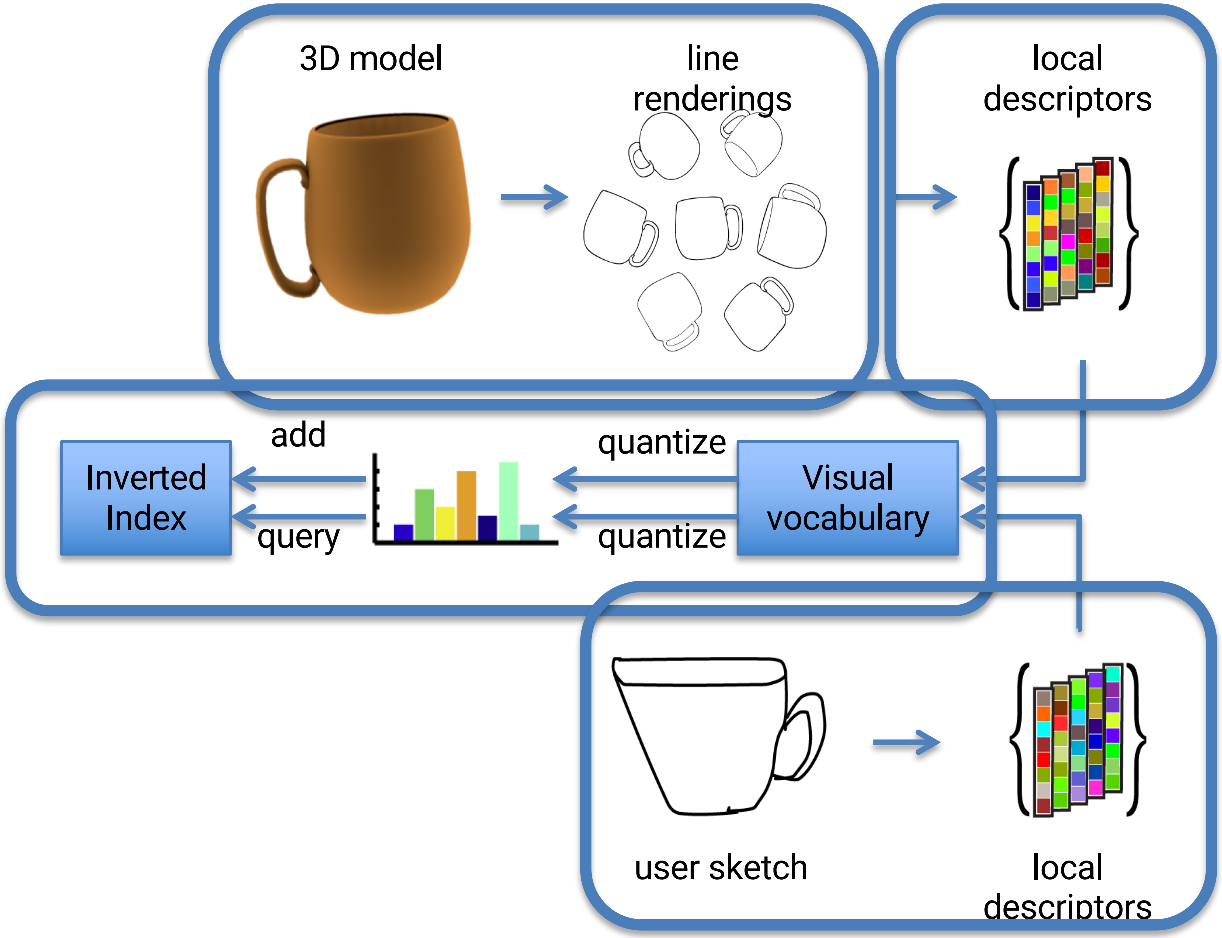
- Tolerate local and global deformations
- Support partial matching
- Fast and efficient

→ Need appropriate feature transform (descriptor)

Overview Computing Pipeline

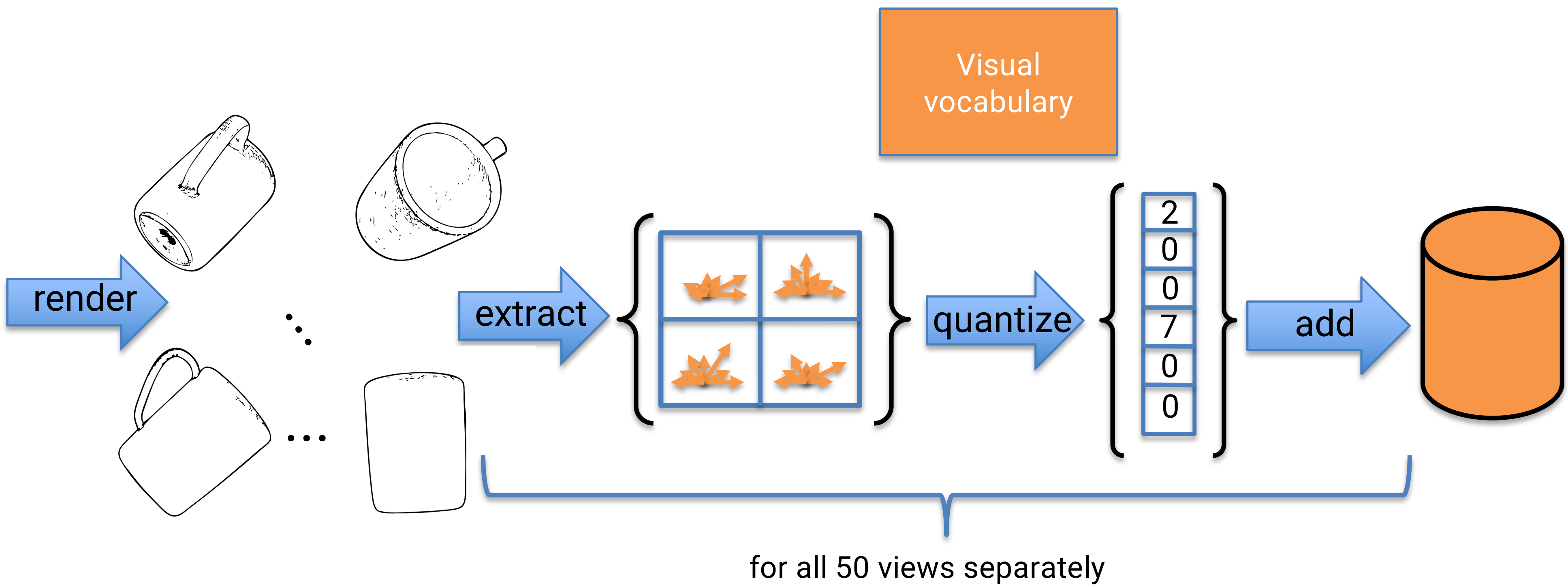
Offline Indexing

Online Search



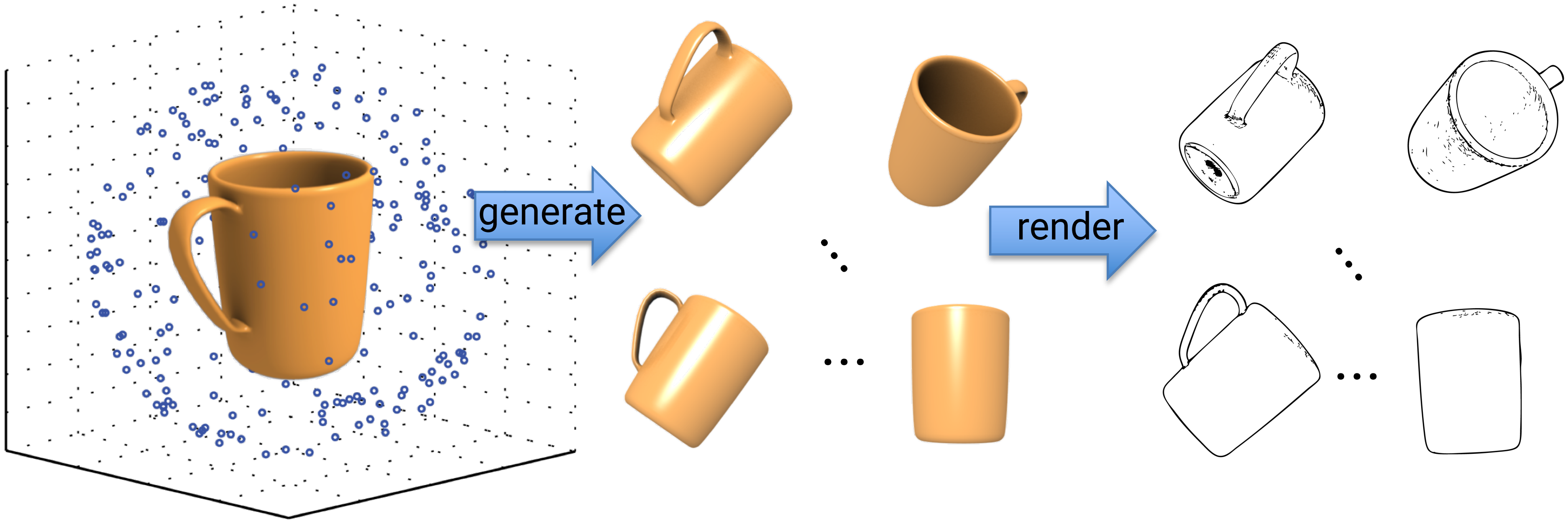
Offline Indexing

Offline indexing



Offline indexing

- Uniformly sample bounding sphere: 50 samples



Feature Transform

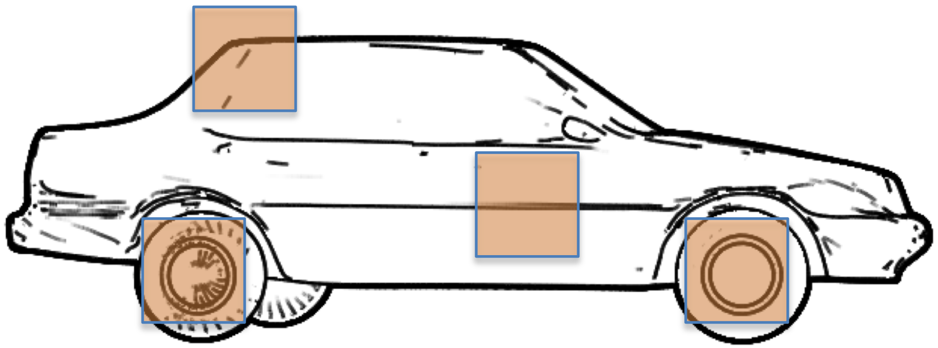
Which feature transforms do we know already?

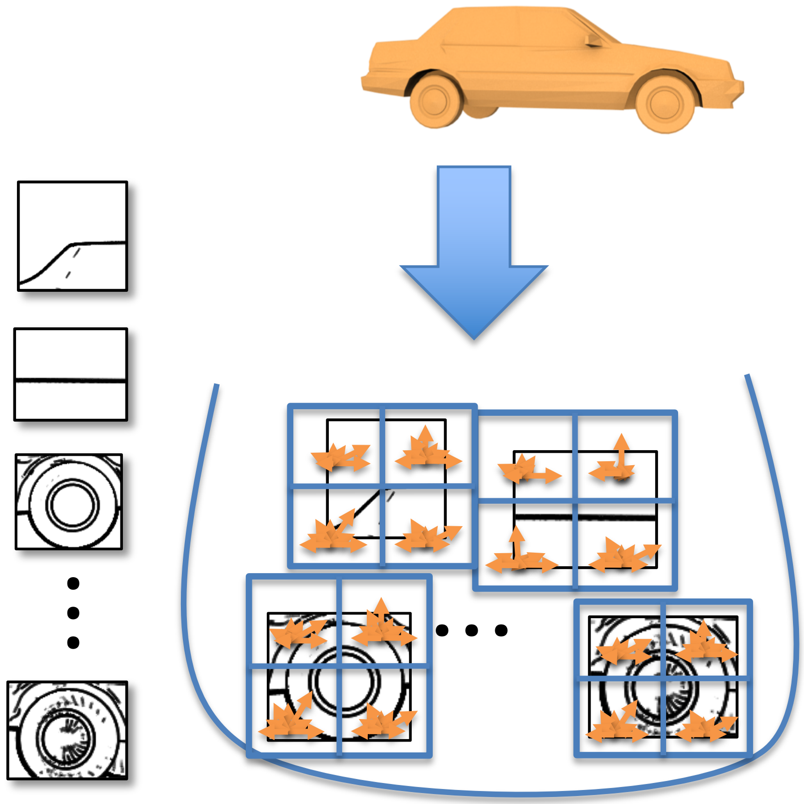
What are their properties?

Where do we extract them?

Local features over all images

- Independent local features allow for:
  - translation invariance
  - partial matching
  - standard search data structures

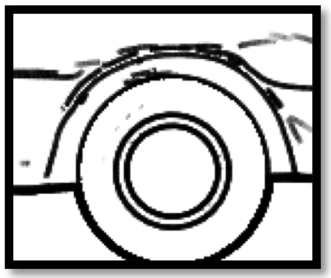




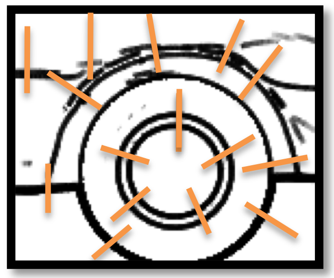
Bag-of-features [Sivic'03]

Offline indexing: features

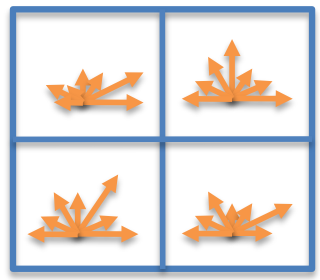
(1) Extract local region



(2) estimate orientations



(3) distribution of orientations

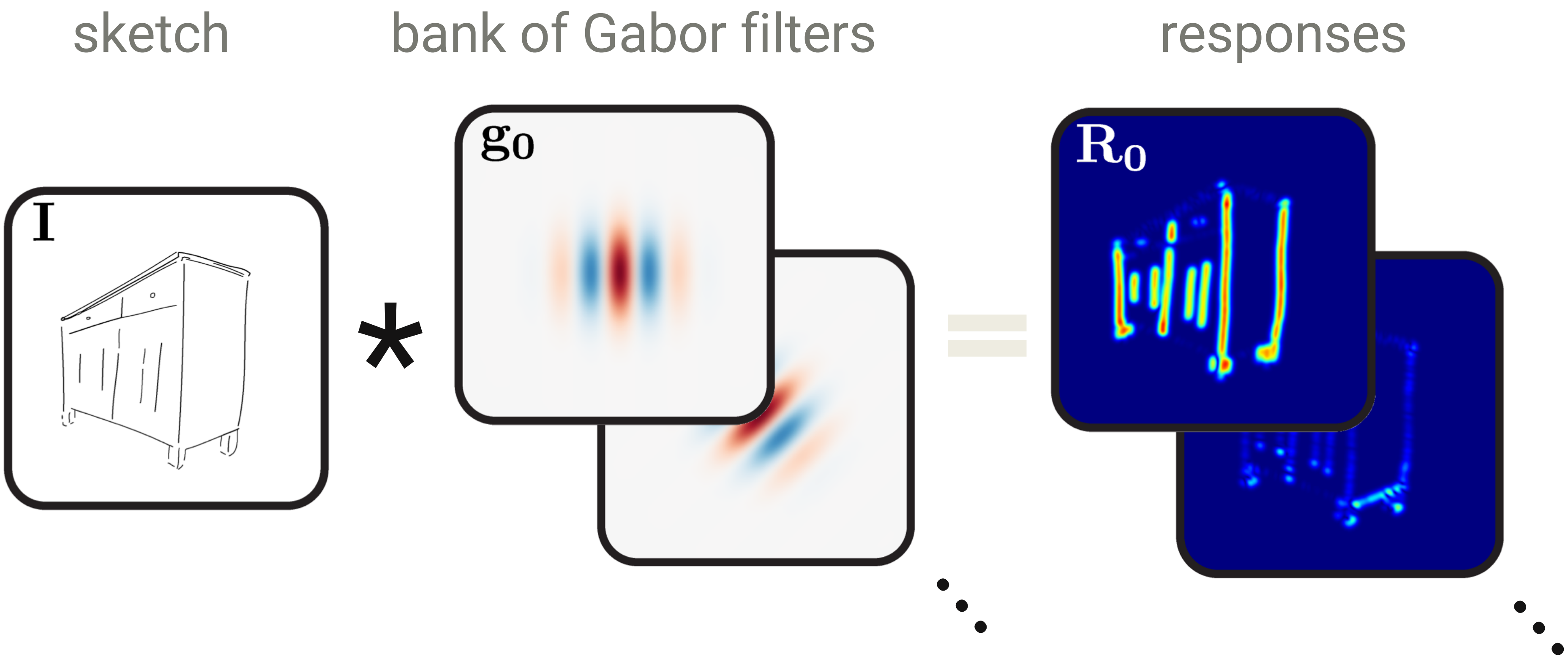


4x4 spatial, 8 radial bins

- No directionality information in gradients
- Binned distribution invariant to small deformations

Feature Transform for Sketches

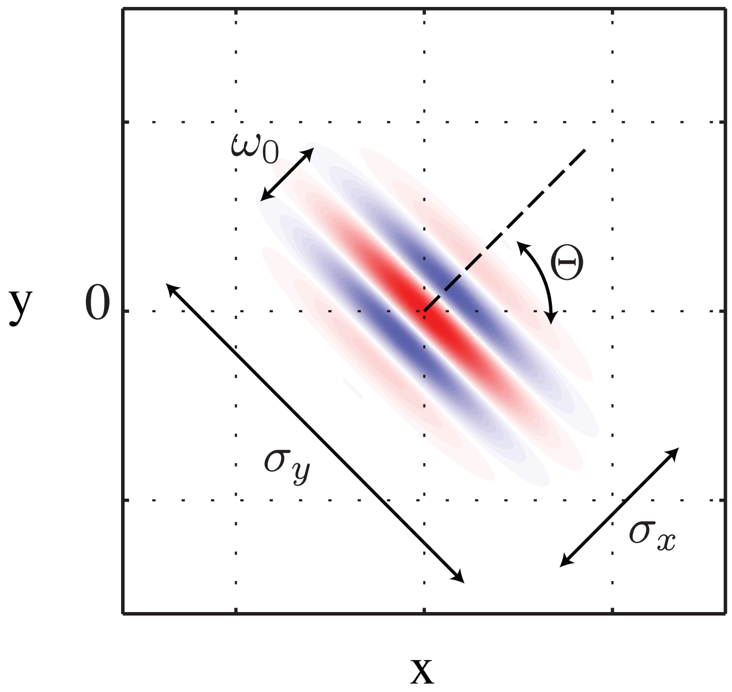
Yet another local Feature Extraction



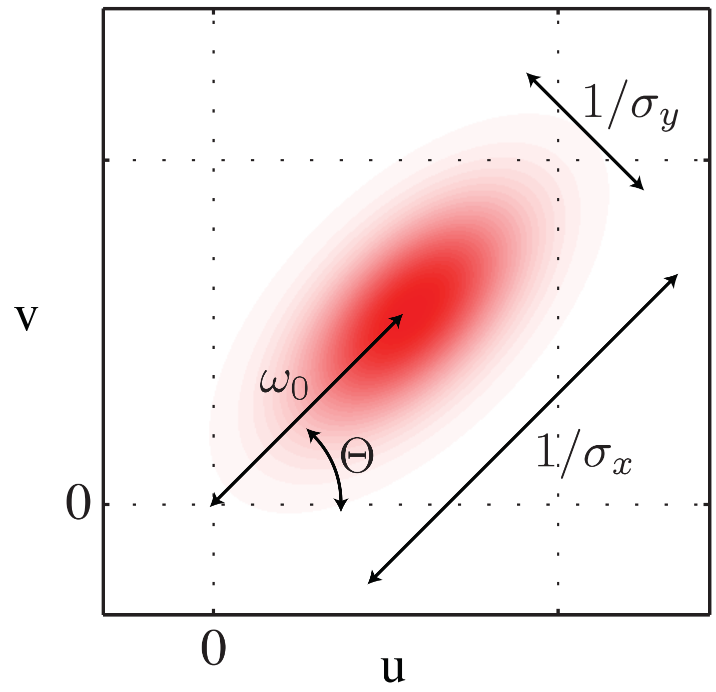
Sketch-based Shape Retrieval. *ACM Transactions on Graphics, Proc. SIGGRAPH 2012.*
Eitz, Mathias, Richter, Ronald, Boubekeur, Tamy, Hildebrand, Kristian and Alexa, Marc.

Local Feature Extraction: Gabor Filter

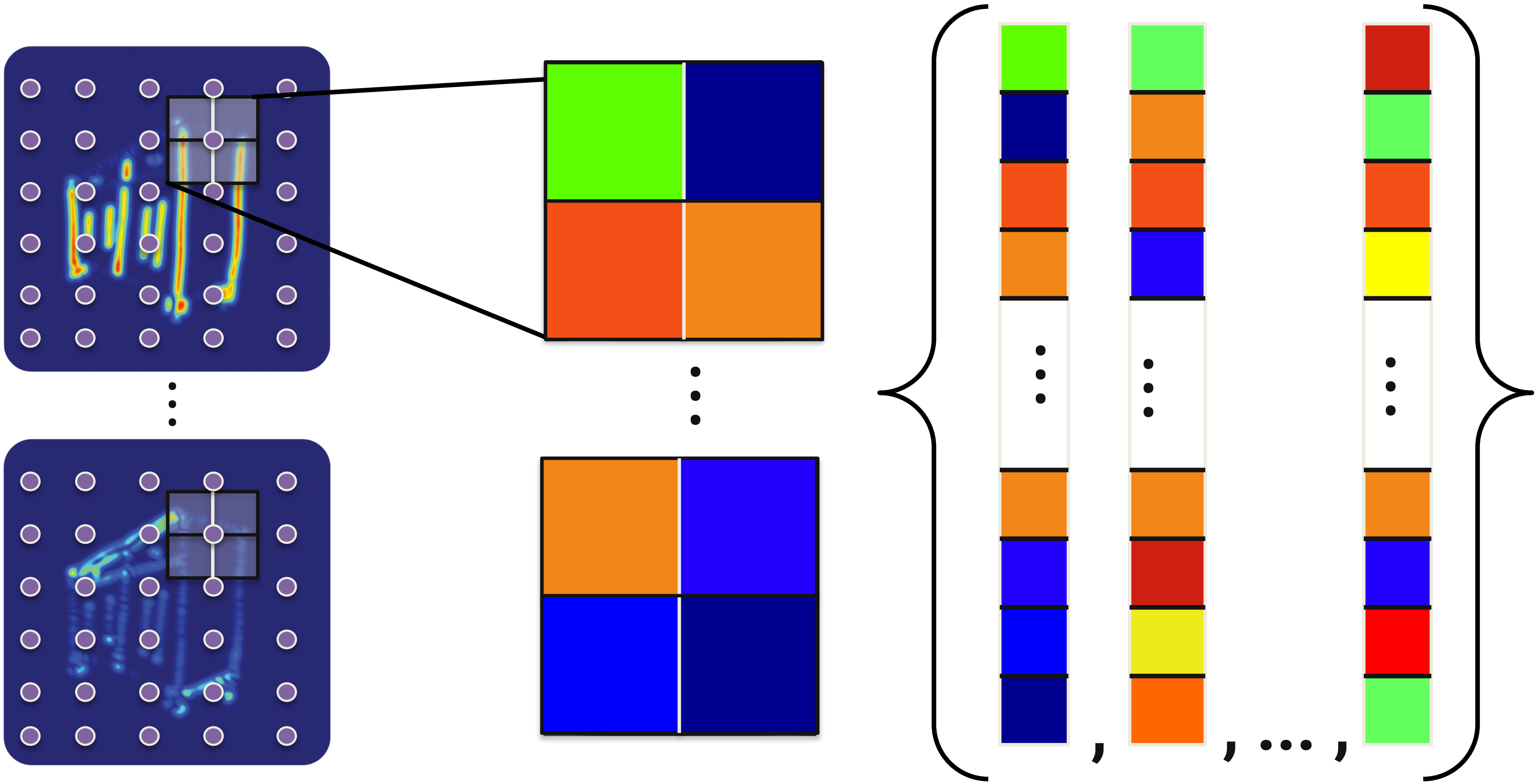
spatial domain



frequency domain

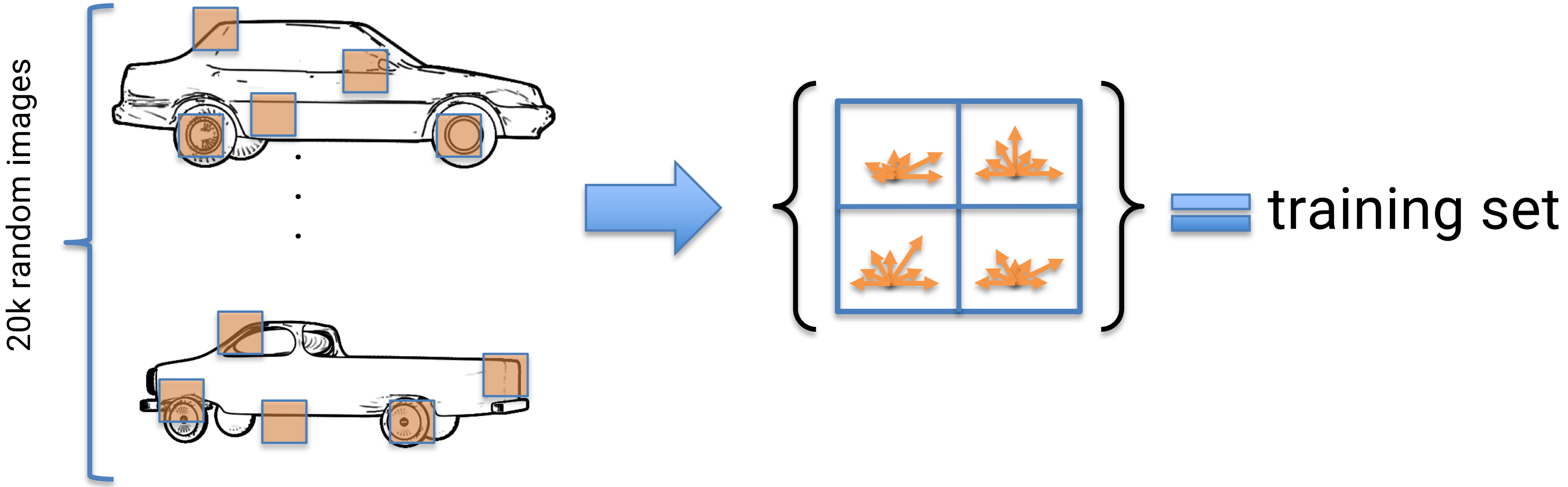


Local Feature Extraction (uniform keypoints)

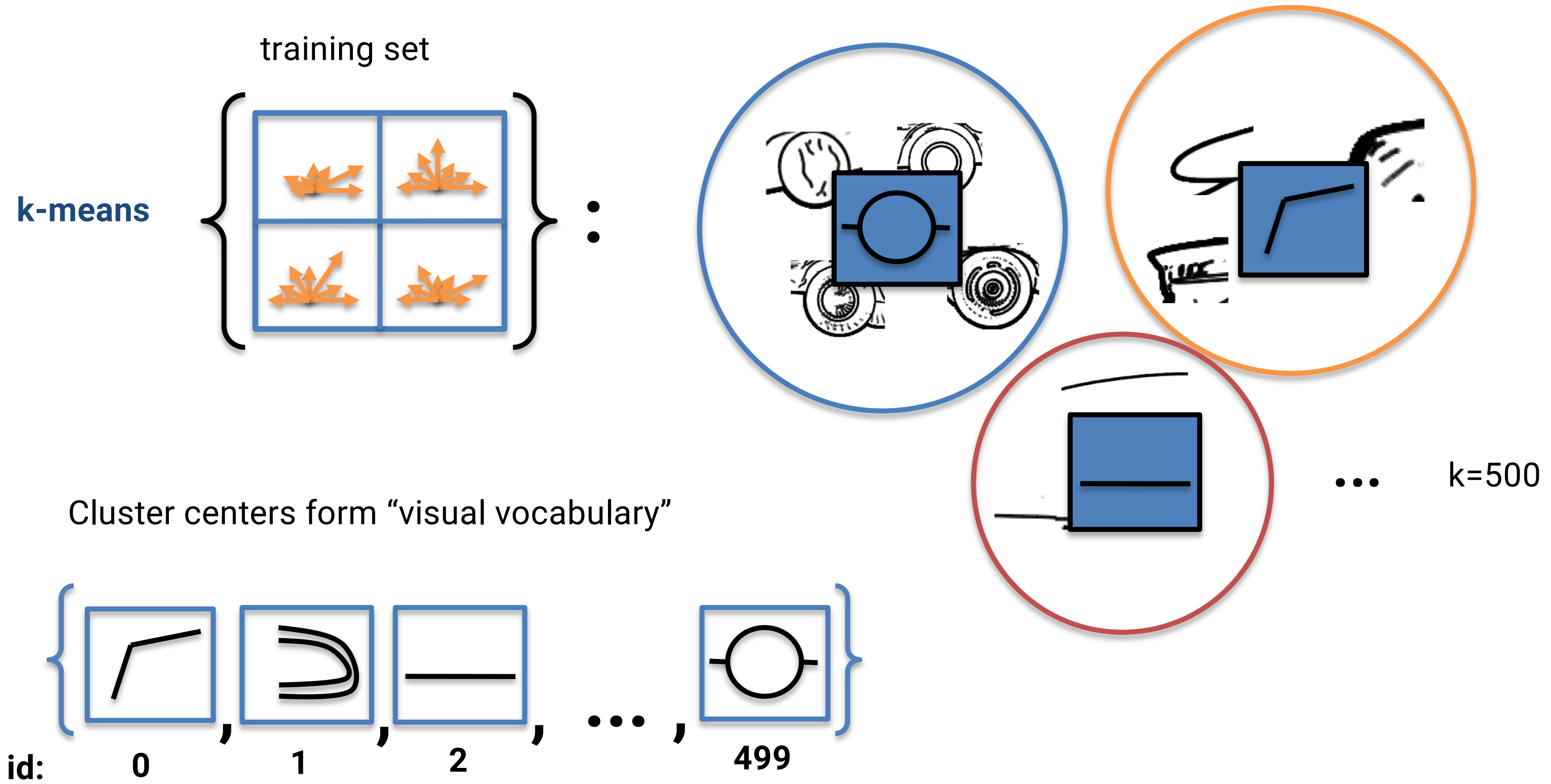


Offline indexing: visual vocabulary

- 20k images (sampled from 50 views each of 2k models)
- 500 local features each
  - Training set size: 10 million local features

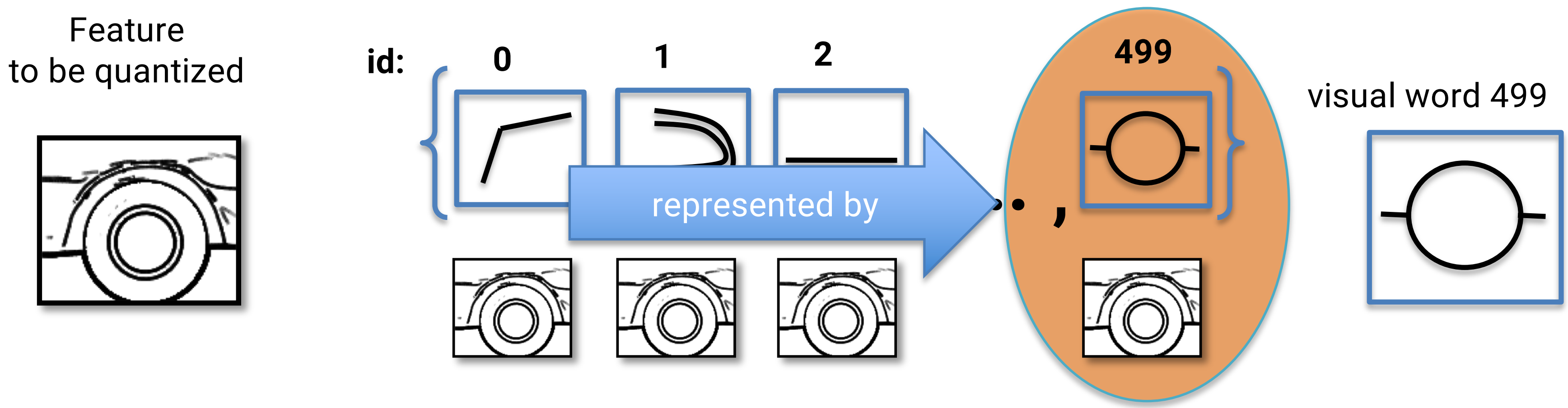


Offline indexing: visual vocabulary

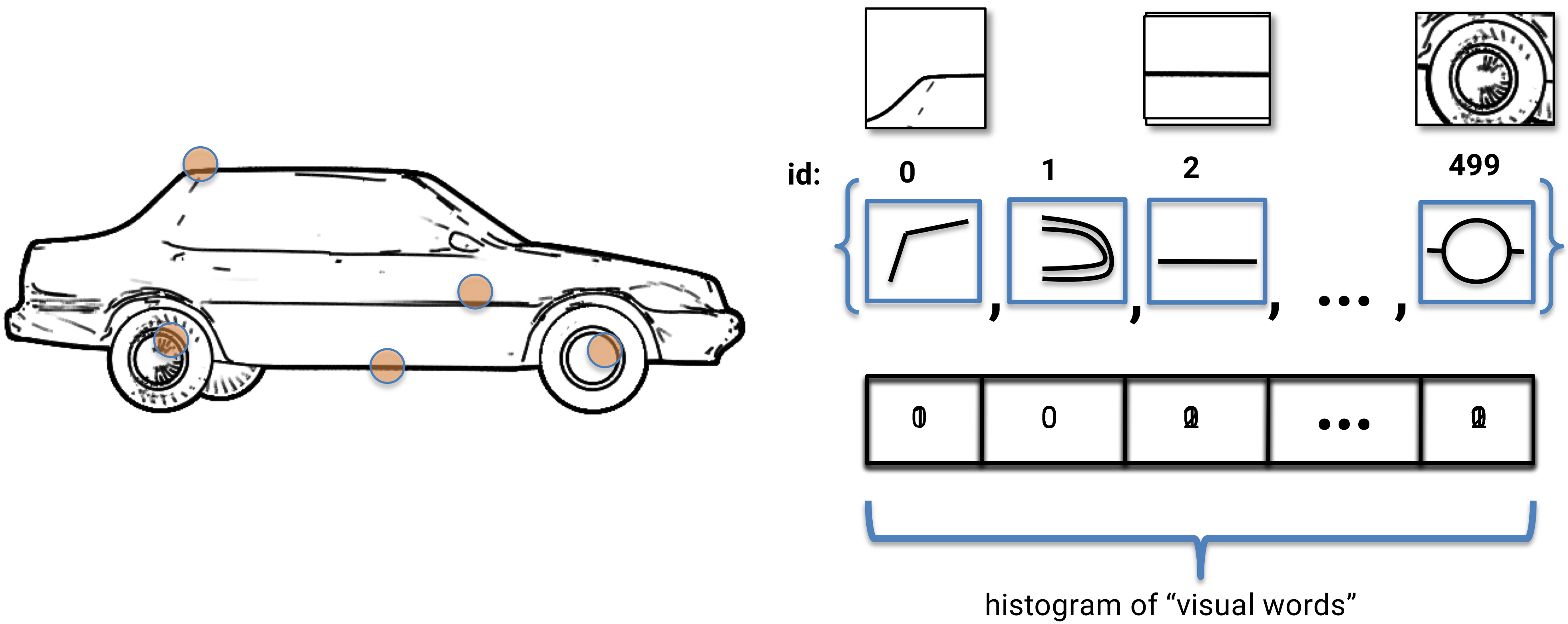


Offline indexing: quantization

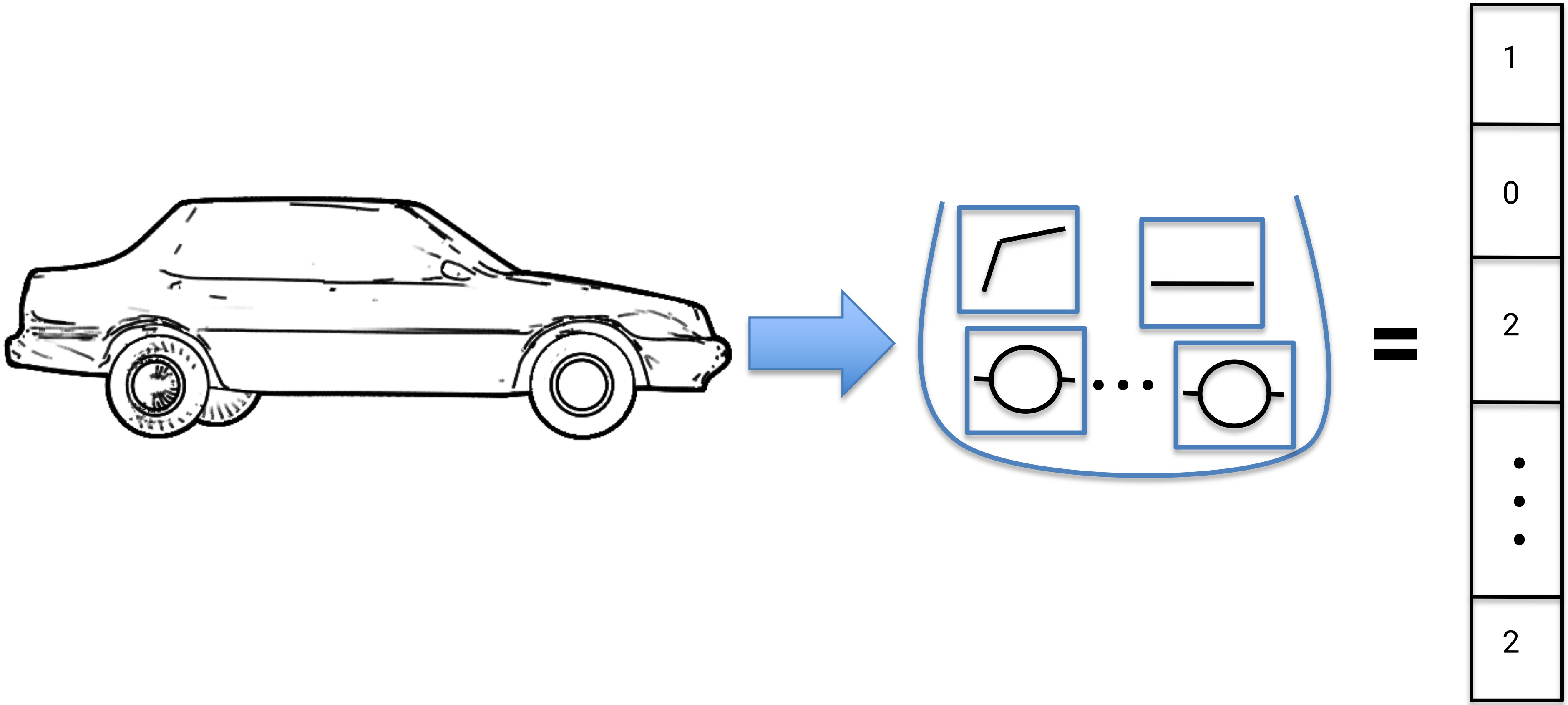
- Quantization allows for
  - More compact representation
  - Grouping of perceptually similar features



Offline indexing: representation



Offline indexing: representation



Inverted Index

Inverted Index – Information Retrieval DS

- Data structure for search engine algorithms
- **Goal:**
  - optimize speed of the query
  - find ‘documents’ (image) where word (descriptor vector) ‘foobar’ occurs
- **Forward index**
  - stores a list of words for each document

Document	Words
doc0	cat, the, world
doc1	cat, has, answer
doc2	cat, world, has

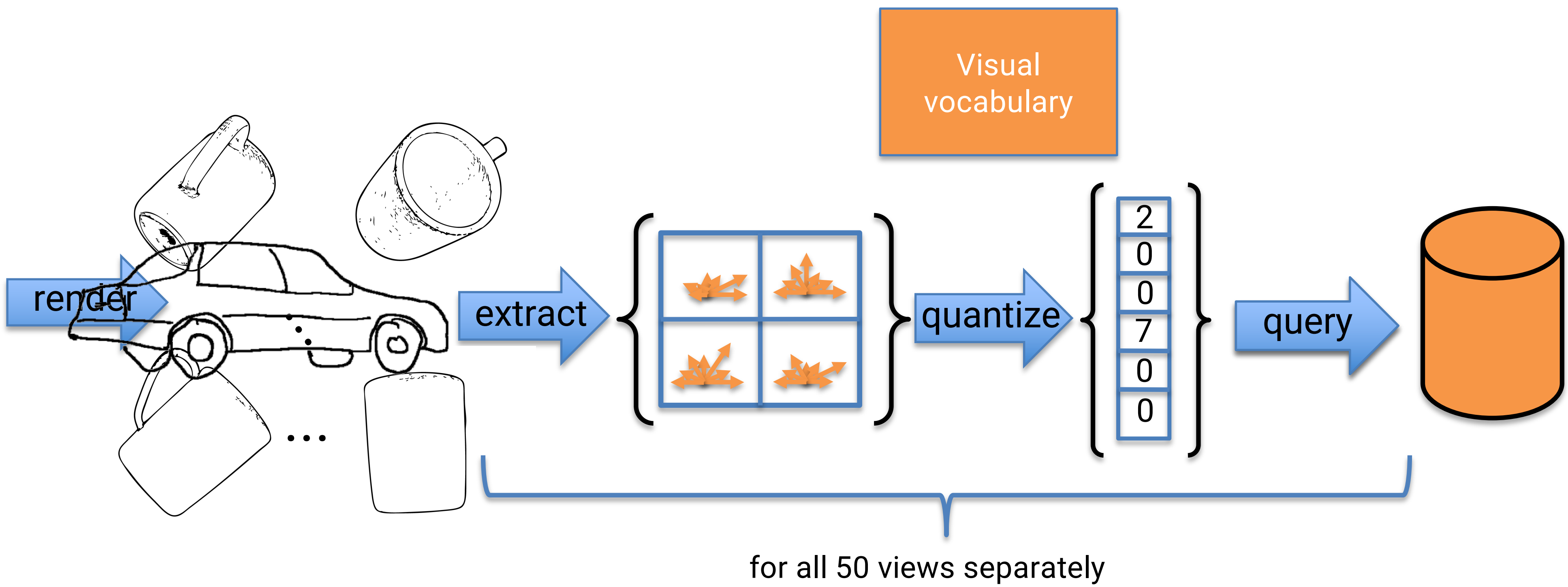
Inverted Index – Information Retrieval DS

- **Querying forward index requires linear search** through all documents + their words (billions)
- Inverted index lists the documents per word

Words	Documents
cat	doc0, doc1, doc2
the	doc0
world	doc0, doc2
has	doc1, doc2
answer	doc1
…	

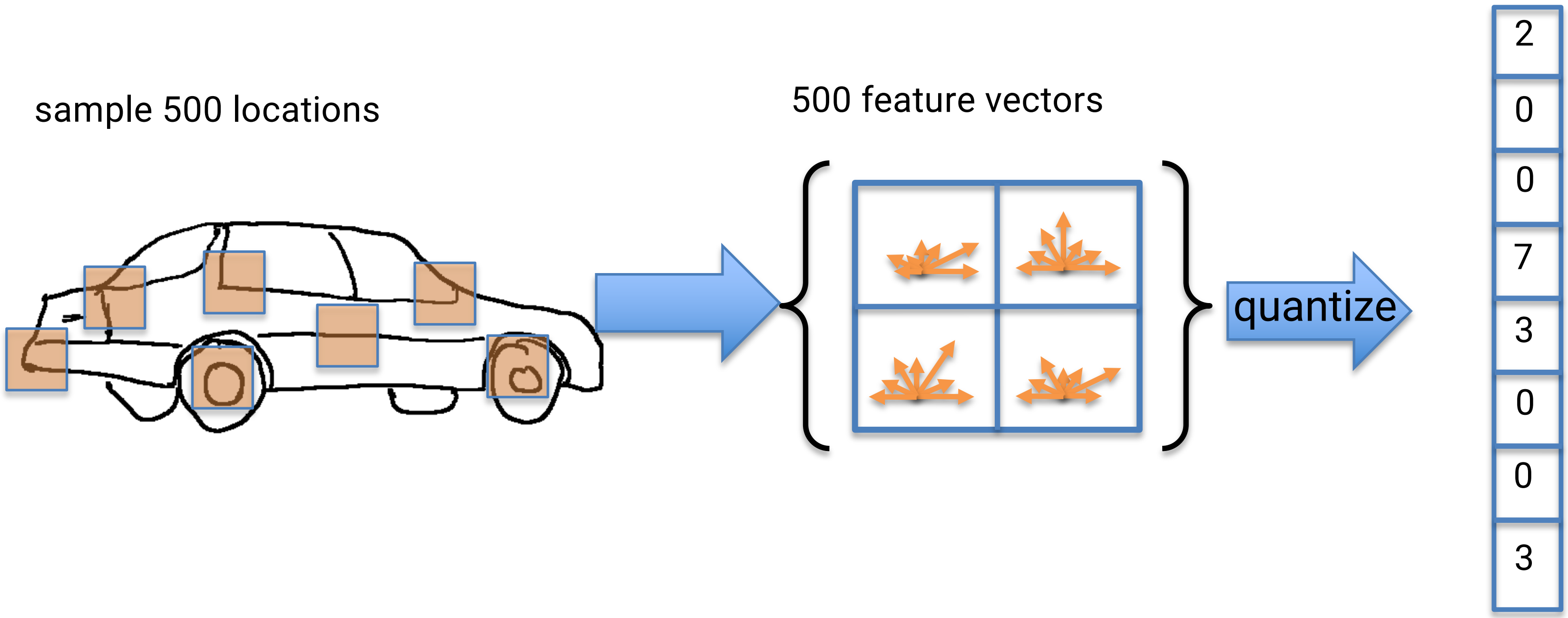
- With the inverted index the query can now be resolved by jumping to the word ID

Online search



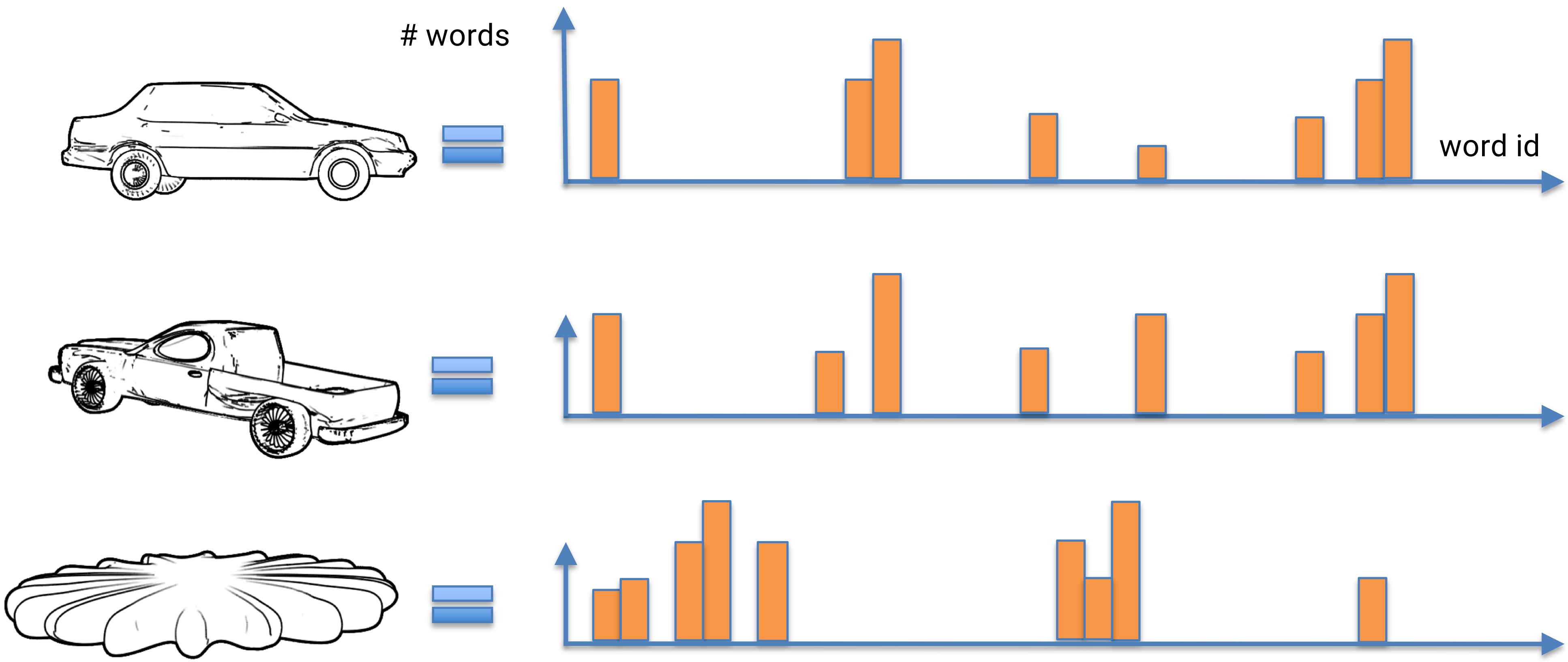
Online Search

Online Search



Online search

- Images as (sparse) histograms of visual words



Tf-idf

- each **document is represented by a vector of word frequencies**
- usual to **apply a weighting** to the components of this vector
- standard weighting is known as **term frequency–inverse document frequency (tf-idf)**

number of occurrences of word *i* in document *d*

k – vector with *k* words

number of documents in database

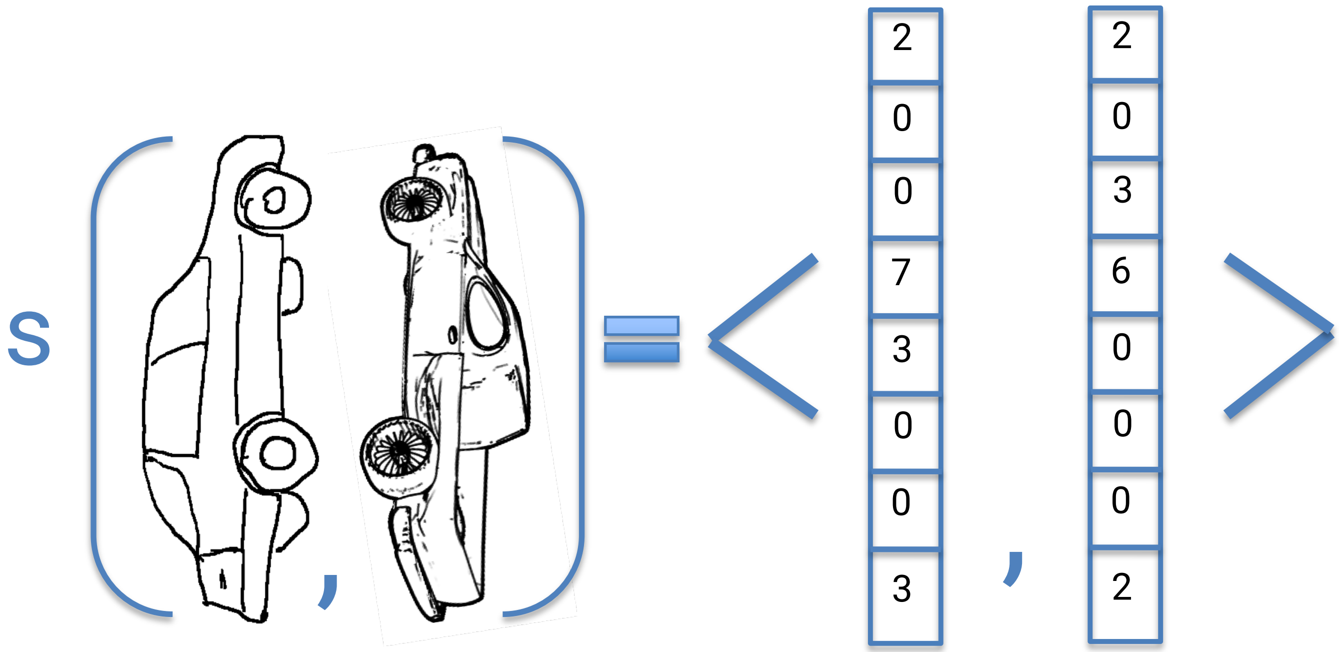
$$t_i = \frac{\boxed{n_{id}}}{\boxed{n_d}} log \frac{\boxed{N}}{\boxed{n_i}}$$

Other weighting schemes possible: e.g. video google weighting

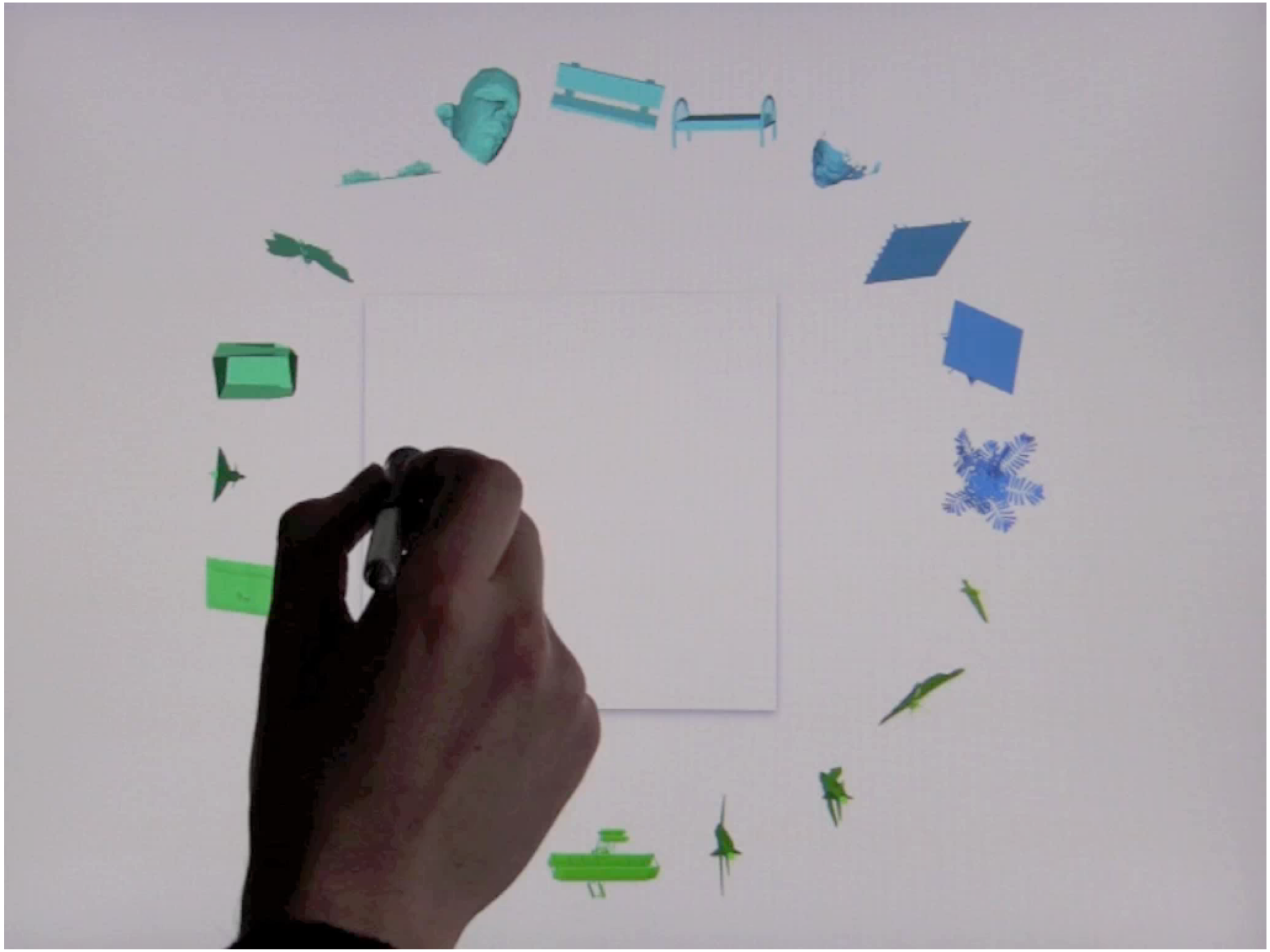
number of occurrences of term i in the whole database

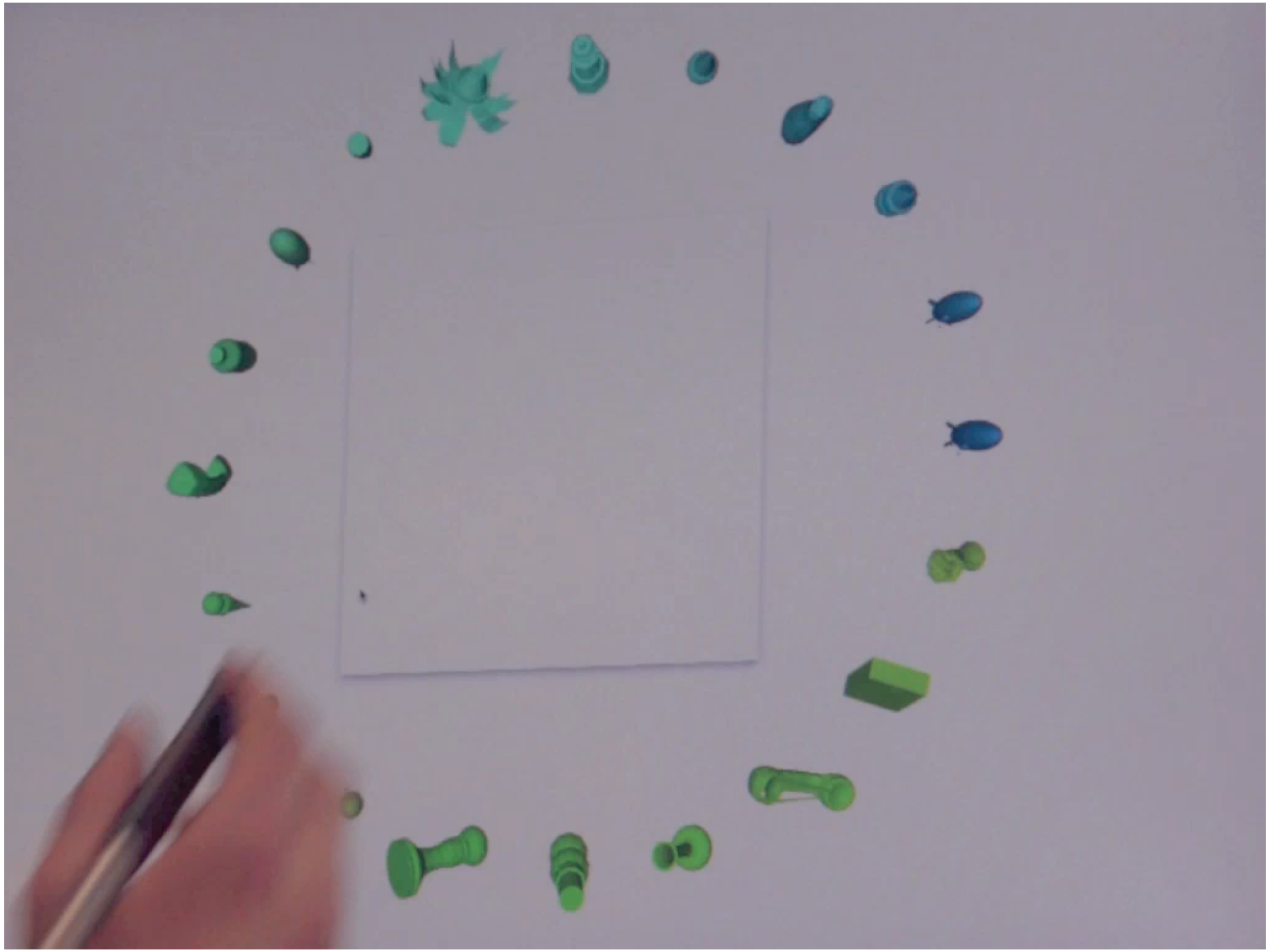
total number of words in document *d*

Online search



- Similarity as angle in high-dimensional space
- Vectors sparse: use inverted index





That’s it for today