Content Based Image Retrieval

- Multimedia data is growing exponentially.
  - Cheap high quality digital imaging devices

  - Sharing of multimedia data on the internet

- Content based organization and retrieval is a viable way of accessing this data.

Why CBIR?

- Historical Achieve
- Forensic documents
- Fingerprint & DNA matching
- Copyright search
- Security usage

Overview

- CBIR has two Approaches:
  - Attribute based
  - Object based
- CBIR can be done by:
  - Color
  - Texture
  - Shape
  - Spatial relationship
  - Semantic primitives
  - Objective Attribute
  - Subjective Attribute
  - Motion
  - Text & domain concepts

Overview

- CBIR has two phases:
    - Database Population phase
        - Image/video shot extraction
        - Feature extraction
    - Retrieval phase
        - Similarity measure

What do users look for?

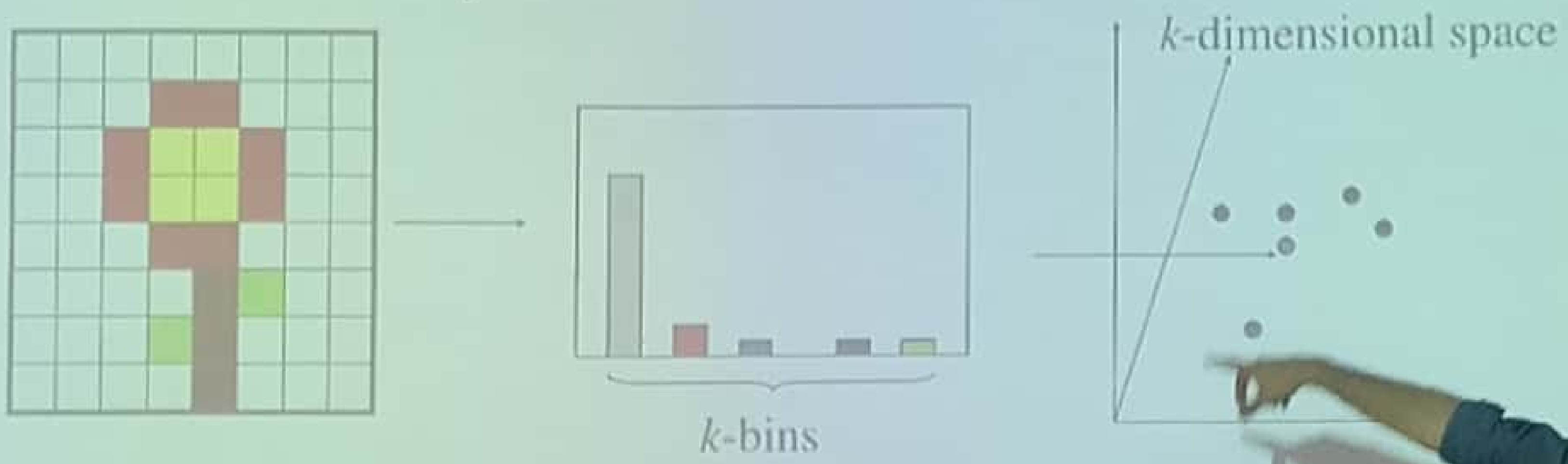
- *In CBIR systems users look for 'things' not 'stuff'.
  - More than global image properties
  - Traditional object recognition will not work
  - Two choices:
    - Rely on text to identify objects
    - Look at regions (objects or parts of objects)

*Chad Carson: Blobworld

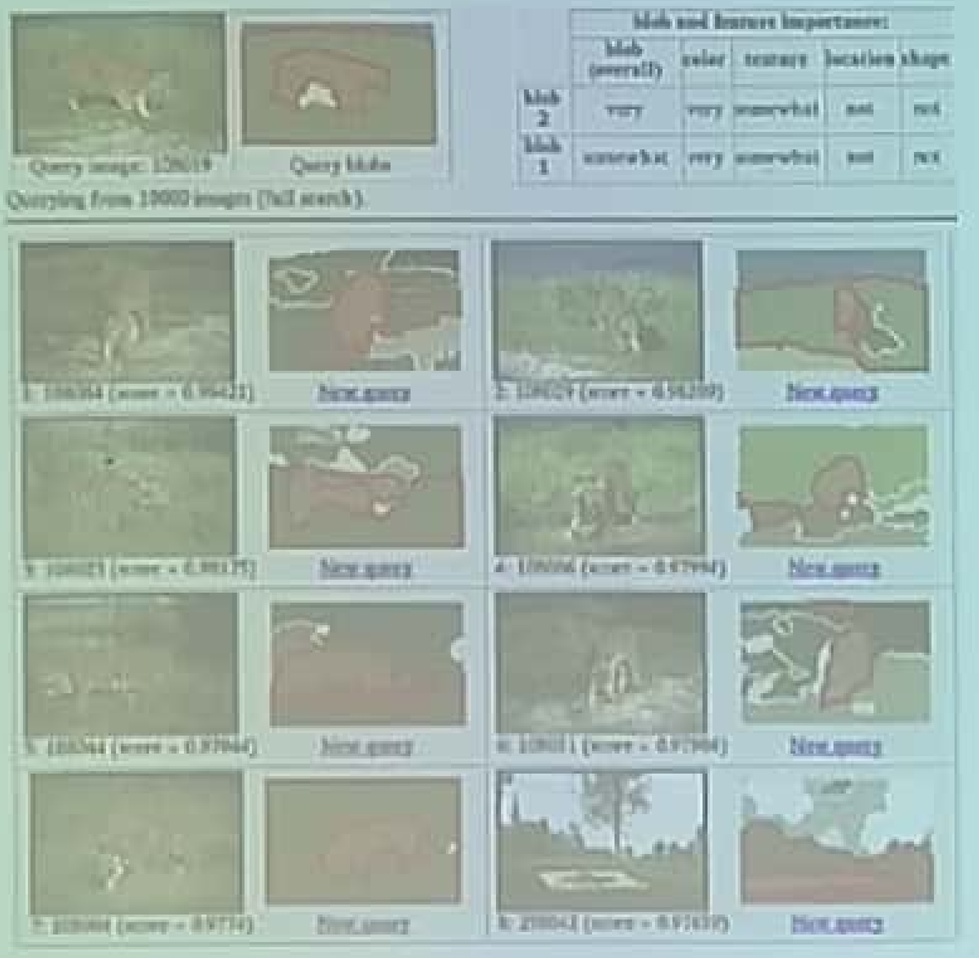
Initial Attempts

- Images represented using simple features
  - Color Histograms
  - Color moments
  - Texture descriptors

- A distance metric (e.g., Euclidean) is used to measure similarity



Initial Successful Systems



Blob World: Carson, Belongie, Malik
CVPR97

SIMPLIcity: Wnag, Li, Wiederhold

Practical CBIR systems

- Practical large scale deployment of CBIR systems require.
  - Efficient indexing and retrieval of thousands of documents.
  - Flexible framework for retrieval based on various types of features. For example Specialized features for highly specific sub domains like faces, vehicles, monuments.
  - Ability to scale up to millions of images without a significant performance trade off.

Lessons From Text Retrieval

- Large scale text retrieval systems have been successfully deployed.
    - Search Engines like

Google bing

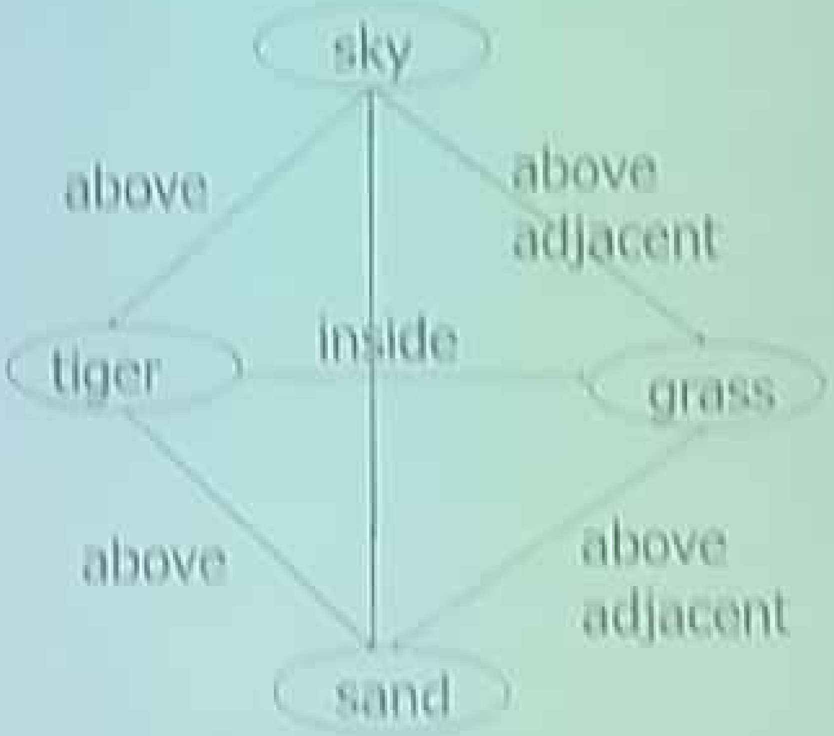
- Efficient indexing and retrieval of millions of documents has been achieved.
- The text retrieval frameworks are adaptive enough to be applied to specialized domains.

Froogle BETA CiteSeer Scientific Literature Digital Library

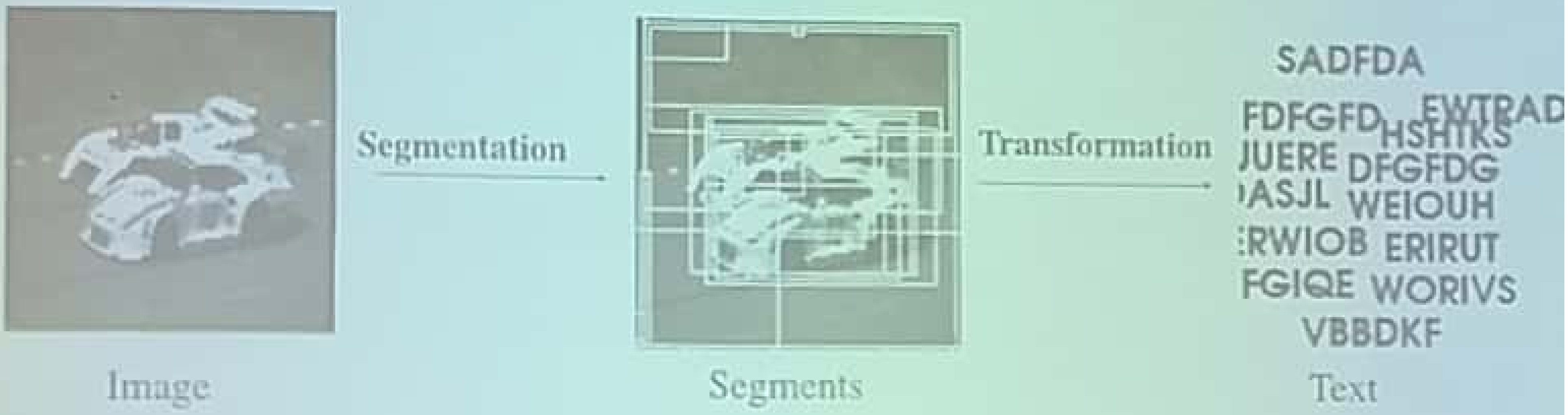
Virtual Graph Representation

image

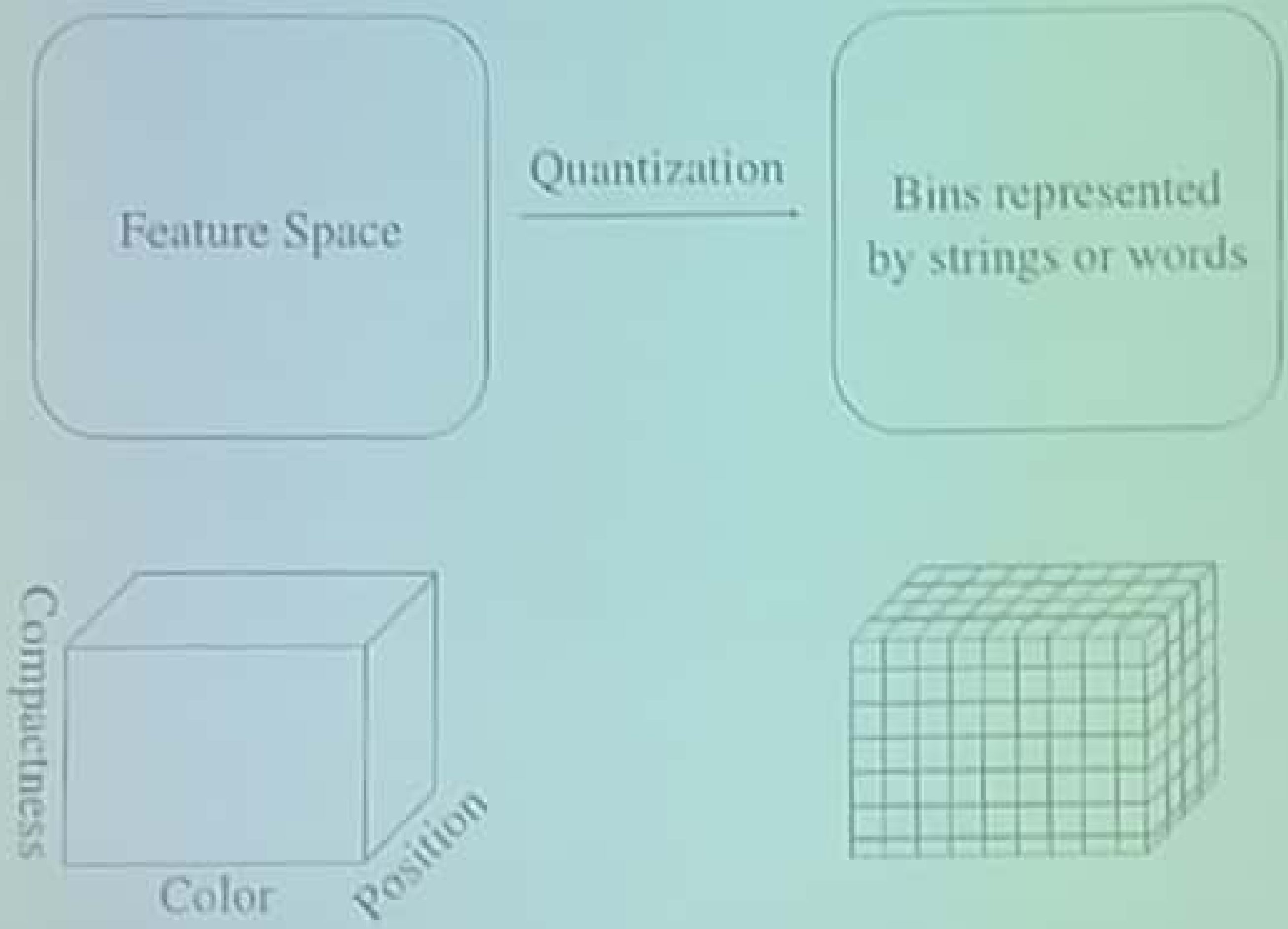
abstract regions



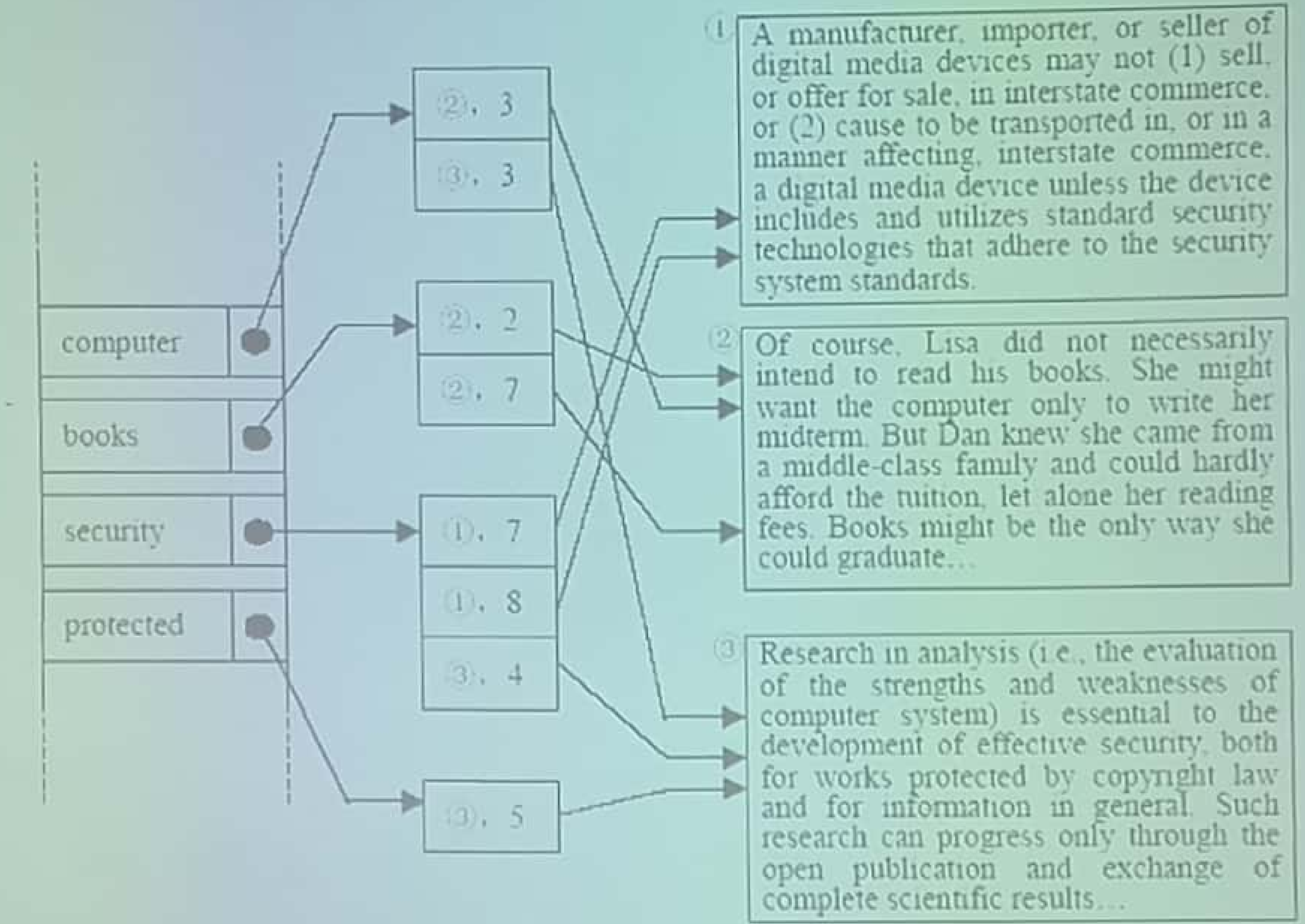
Virtual Textual Representation



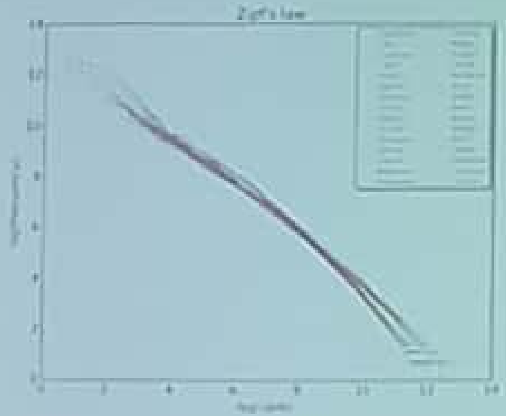
Transformation



Inverted Index



Assigning Weights to Terms



A plot of the rank versus frequency for the first 10 million words in 30 Wikipedias (dumps from October 2015) in a log-log scale.

- Binary Weights
- Raw term frequency
- tf x idf
  - Recall the Zipf distribution
  - Want to weight terms highly if they are
    - frequent in relevant documents ... BUT
    - infrequent in the collection as a whole

Zipf's law states that given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the frequency table. Thus the most frequent word will occur approximately twice as often as the second most frequent word, three times as often as the third most frequent word, etc. - Wiki

Why IDF?

- IDF provides high values for rare words and low values for common words:

For a collection of 10000 documents:

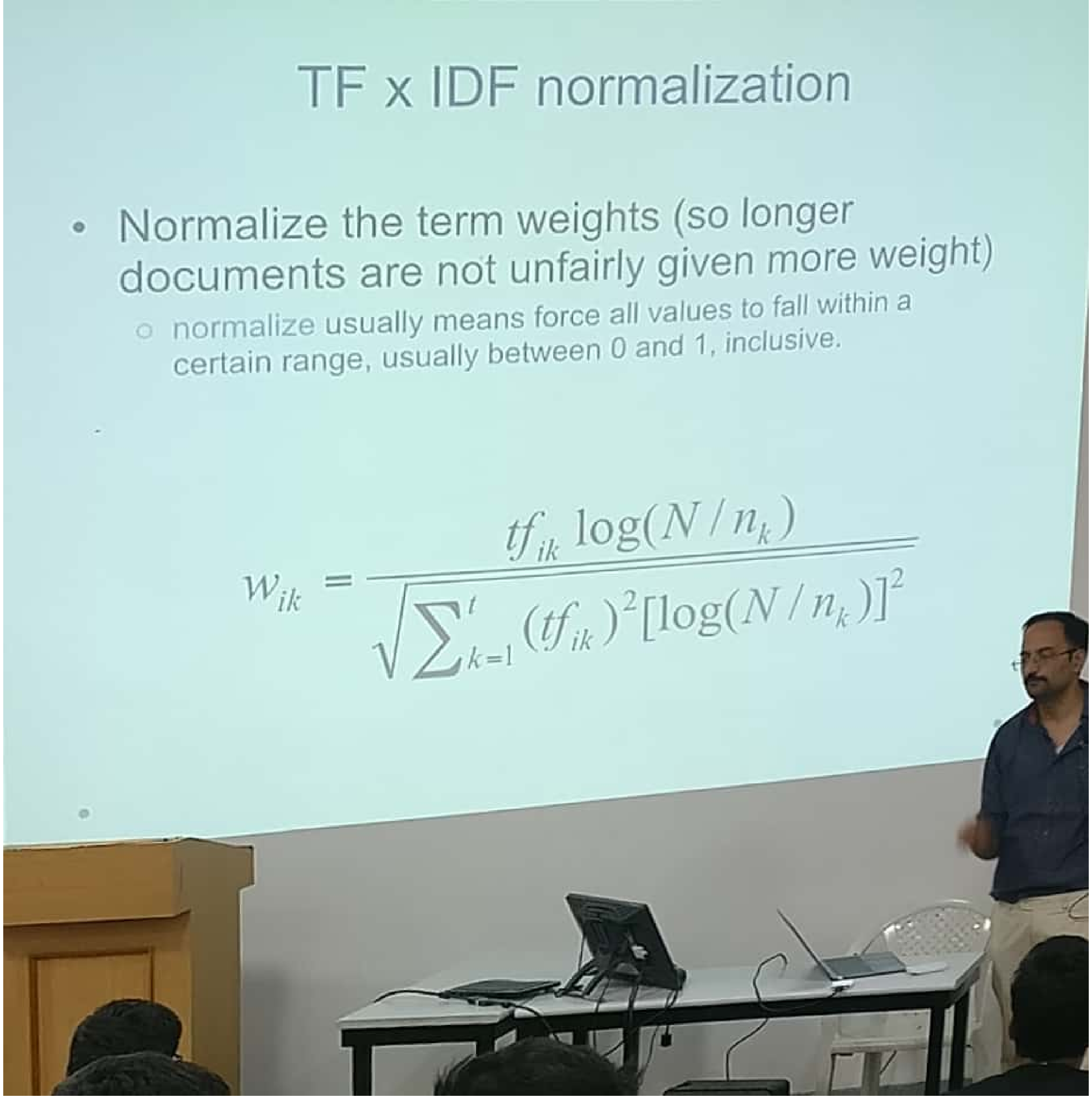
$$\log\left(\frac{10000}{10000}\right) = 0$$

$$\log\left(\frac{10000}{5000}\right) = 0.301$$

$$\log\left(\frac{10000}{20}\right) = 2.698$$

$$\log\left(\frac{10000}{1}\right) = 4$$

- Useful for Ranking


TF x IDF normalization
• Normalize the term weights (so longer documents are not unfairly given more weight)
○ normalize usually means force all values to fall within a certain range, usually between 0 and 1, inclusive.
$$w_{ik} = \frac{tf_{ik} \log(N / n_k)}{\sqrt{\sum_{k=1}^{t} (tf_{ik})^2 [\log(N / n_k)]^2}}$$