Navigator
— Similarity Based Browsing for Image & Video Databases —

Damian Borth, Christian Schulze,
Adrian Ulges, Thomas M. Breuel

Image Understanding and Pattern Recognition
DFKI & TU Kaiserslautern

Motivation

How to retrieve & browse in large databases of images or videos?

▶ provide global view
   → browse through all images of last year

▶ find visual similar images
   → find all pictures with a beach

▶ Movie scene finder
   → find scenes of Batcave form “The Dark Knight”

▶ TV network archives
   → need a desert video clip for news report
Browsing Metaphor: Stack of images

Start Browsing

- no query is selected
- a random set of images is displayed
- user selects image which roughly represents this query
Browsing Metaphor: Stack of images

Zooming & Panning

▶ zooms into the database i.r.t. selected query
▶ displays a more detailed cluster of images
▶ panning = refinement of query
▶ further zooms will lead to different results
Related Work

Query-by-Example [Flickner95]
- database is searched according to given image
- tend to provide a localized view of the database
- How to get an overview?

ViBE System [Chen04]
- browsing based on Similarity Pyramids
- browsing performed only at shoot level
  → shot = one keyframe
  → loosing visual details
Our Approach

System Pipeline

- Keyframe Extraction
  → only for video databases
- Feature Extraction
- Balanced Hierarchical Clustering
- User Interface
Keyframe Extraction

Notation

- video database
  \[ DB_{\text{video}} = X_1, \ldots, X_n \]

- every video \( X_i \in DB_{\text{video}} \) is represented by a set of \( \{ x_{i1}, \ldots, x_{im} \} \) keyframes

\[ \rightarrow \text{resulting in a total set of } \{ x_1, \ldots, x_k \} \text{ keyframes for the entire } DB_{\text{video}} \]
Keyframe Extraction

Two Step Method

- Shot Boundary Detection
  - Color Layout Desc. (CLD) differences
  - Adaptive Thresholding
    [Lienhart01]

- Intra-Shot Clustering
  [Hammoud00]
  - CLD feature space
  - K-Means
  - Number of clusters: Bayesian Information Criterion (BIC)
Feature Extraction & Hierarchical Clustering

**Feature Extraction**

- A feature vector \( z_j \in \mathbb{R}^D \) is extracted for every keyframe \( x_j \in \{x_1, \ldots, x_k\} \)

- \( z_j \in \mathbb{R}^D \) consists of equally weighted color & texture features
  - Color Histograms - 8 × 8 × 8 binning of RGB Cube
  - Tamura Texture Histograms [Tamura78]
Feature Extraction & Hierarchical Clustering

Feature Extraction

▶ a feature vector $z_j \in \mathbb{R}^D$ is extracted for every keyframe $x_j \in \{x_1, ..., x_k\}$

▶ $z_j \in \mathbb{R}^D$ consists of equally weighted color & texture features
  ▶ Color Histograms - $8 \times 8 \times 8$ binning of RGB Cube
  ▶ Tamura Texture Histograms [Tamura78]

Clustering Overview

(a) Feature Space of $z_j \in \mathbb{R}^D$
(b) Clustering Result
(c) Dendrogram
(d) Generated Binary Tree
Agglomerative Hierarchical Clustering

1. **Initialization**: - each feature vector \( z_j \in \{ z_1 \ldots z_k \} \) represents a *singleton* cluster \( c_j \in \{ c_1 \ldots c_k \} \)
   - compute Euclidean distance for singletons

2. merge two nearest clusters: \( \{ c_i \} \land \{ c_j \} \rightarrow \{ c_i, c_j \} \)

3. update distances i.r.t. merged clusters

4. go to step 2 until \( |c| = 1 \)
Agglomerative Hierarchical Clustering

1. **initialization**: - each feature vector $z_j \in \{z_1...z_k\}$ represents a *singleton* cluster $c_j \in \{c_1...c_k\}$
   - compute Euclidean distance for singletons

2. merge two *nearest* clusters: $\{c_i\} \land \{c_j\} \rightarrow \{c_i, c_j\}$
3. update distances i.r.t. merged clusters
4. **go to step 2 until** $|c| = 1$

**Nearest = Used Linkage Method**
- Singelton Linkage
- Complete Linkage
- Average Linkage
Balanced Hierarchical Clustering (cont)

Average Linkage

\[ d(c_i, c_j) = \frac{1}{|c_i| \times |c_j|} \times \sum_{z_i \in c_i} \sum_{z_j \in c_j} d(z_i, z_j) \]

- distance between “means” of cluster pairs
Balanced Hierarchical Clustering (cont)

**Average Linkage**

\[ d(c_i, c_j) = \frac{1}{|c_i| \cdot |c_j|} \sum_{z_i \in c_i} \sum_{z_j \in c_j} d(z_i, z_j) \]

- distance between “means” of cluster pairs

**Clustering Result**

- binary tree is linearized → not usable for browsing
Balanced Hierarchical Clustering (cont)

**Modified Average Linkage**

\[ d(c_i, c_j) = \frac{1}{|c_i| \times |c_j|} \times \sum_{z_i \in c_i} \sum_{z_j \in c_j} d(z_i, z_j) + \alpha \times (|c_i| + |c_j|) \]

- Balancing Term: \( \alpha \times (|c_i| + |c_j|) \), where \( 0 \leq \alpha \leq \infty \)
- empirically gave best results with \( \alpha = 0.01 \)
Balanced Hierarchical Clustering (cont)

Modified Average Linkage

\[ d(c_i, c_j) = \frac{1}{|c_i| \cdot |c_j|} \cdot \sum_{z_i \in c_i} \sum_{z_j \in c_j} d(z_i, z_j) + \alpha \cdot (|c_i| + |c_j|) \]

- Balancing Term: \( \alpha \cdot (|c_i| + |c_j|) \), where \( 0 \leq \alpha \leq \infty \)
- empirically gave best results with \( \alpha = 0.01 \)

Clustering with Balancing Term

- balancing term forces clustering to prefer small clusters
- binary tree is more balanced → preferred browsing structure
Balanced Hierarchical Clustering (tree building)

Binary Tree Generation

- postprocessing of dendrogram structure into a binary tree
- neglecting similarity measurement given by dendrogram
- singleton clusters $c_i$ define leafs of binary tree
- merging points \{ $c_i, c_j$ \} define inner nodes of binary tree
Question: How to arrange keyframes within a cluster?

- Elements within a cluster do not have a spatial arrangement
- Additional processing to provide spatial organization → *Quad-Tree to Pyramid Mapping* [Chen98]
- Our Approach: Use the already available cluster hierarchy → Inorder Traversal of tree structure
Navidgator User Interface
Navidgator User Interface

User Interface

1. Selected Query
2. Similarity Cluster (if query refinement is needed)
3. Navigation Menu
4. Browser History
5. Restart Browsing
Live Demo

...do we have Internet access here?...

(demo also available at the DFKI-IUPR booth)
Summary

Typical Browsing Scenario

▶ annotation (tags) are not always available
▶ user has a rough visual concept of the query in mind

Navigator Prototype

▶ browsing with “query by example” paradigm
▶ use of dynamic query refinement
▶ allows to zoom into different levels of detail

Further Work

▶ dealing with growing databases
  ▶ online clustering
  ▶ tree merging
References


