11.1 Video Similarity

- **Similarity** is important:
  - Ranking of the retrieval results
  - Finding duplicates (different resolution, coding, etc.)
  - Detecting copyright infringements
- Various measures for the similarity
  - Simple idea: percentage of frames with high visual similarity
    - Analogous to Tanimoto similarity measure for texts: percentage of identical words in two texts (relative to the total number of words)

- Fundamental step is the identification of (audio) **visual features** from the frames (time series of features)
  - Color distribution, motion, etc.
- For **efficiency reasons**, the similarity should not be determined between frames, but between shots

11.1 Video Similarity

- We usually have to consider…
  - The higher the **number of features**, the more properties can be used in the similarity measure (i.e. similarity measures get more accurate), but the more **inefficient** is the retrieval process
  - In general, for videos the **accuracy** of the scoring is not the critical factor, but **efficiency** is very important

- 65,000 videos uploaded each day on YouTube
  - Prone to duplicates
  - Redundancy is severely hampering video search
  - Eliminate duplicates
  - What are duplicates?
11.1 Video Similarity

- For identical copies it's easy! But... we have to deal with "near duplicates"  
  - See e.g., (Wu, Ngu and Hauptmann, 2006)
- Near-duplicate web videos are "essentially the same", differing in:  
  - File formats  
  - Encoding parameters  
  - Photometric variations (color, lighting changes)  
  - Editing operations (caption, logo and border insertion)  
  - Different lengths

11.1 Video Similarity

- "The lion sleeps tonight"

11.1 Similarity Measures

- Assumptions  
  - Each frame is represented through a (high dimensional) feature vector in a metric space \( F \) with distance measure (metric) \( d \)  
  - The similarity measure (for videos) is invariant with respect to the shot sequence  
- Thus,...  
  - Representation of videos by finite (unordered) sets of feature vectors

11.1 Similarity Measures

- \( d(x, y) \) is the distance (dissimilarity) between two feature vectors \( x \) and \( y \)  
- Vectors (represented by frames) \( x \) and \( y \) are visually similar, if \( d(x, y) \leq \varepsilon \) for \( \varepsilon > 0 \) (independent of the actual values of \( x \) and \( y \))  
  - Approach after Cheung and Zakhor, 2003
11.1 Video Similarity

- Basic idea: compute the percentage of similar frames in the videos
  - Naive video similarity: the total number of frames of a video, which are similar to at least one frame in the other video, divided by the total number of frames

11.1 Video Similarity

\[ nvs(X, Y; \varepsilon) = \frac{\sum_{x \in X} 1_{\exists y \in Y: d(x, y) \leq \varepsilon}}{|X|} + \frac{\sum_{y \in Y} 1_{\exists x \in X: d(x, y) \leq \varepsilon}}{|Y|} \]

- Indicator function \( 1_A \) for a set \( A \): value of 1 if \( A \) is not empty, value 0 otherwise
- If each frame in \( X \) can be mapped in a similar frame in \( Y \) (and vice versa), \( nvs = 1 \)
- \( nvs = 0 \), if there are no similar frames in the two videos

11.1 Video Similarity

- Naive video similarity is often not intuitive
  - Shots may contain many visually similar frames
  - The specific number within one shots, depend on the exact encoding, and should therefore not influence the measure
  - E.g., generate \( Y \) through multiplication of a single frame from \( X \). For \( |Y| >> |X| \) \( nvs(X, Y; \varepsilon) \approx 1 \)

11.1 Video Similarity

- E.g., frames of video \( X \) are marked with ", frames of video \( Y \) with ".
- Then the "intuitive" distance is about 0.5, the calculated one is however, 0.9

11.1 Video Similarity

- Solution: consider quantities of similar frames as fundamental units
  - Without regarding the temporal structure (representation as a set of feature vectors) we combine all visually similar frames to clusters
  - Two frames \( x, y \in X \) belong to the same cluster if \( d(x, y) \leq \varepsilon \)
  - Problem: consistent cutting is not always possible
    - If \( d(x, y) \leq \varepsilon \) and \( d(y, z) \leq \varepsilon \), then what is with \( d(x, z) \)?

- In single link clustering, \( d(x, y) \leq \varepsilon \) implies that \( x \) and \( y \) are in the same cluster, not vice versa
  - The clusters \( [X]_\varepsilon \) of a video \( X \) are the connected components in "distance < \varepsilon"-graph
  - A cluster is called \( \varepsilon \)-compact if all the frames of the cluster have at most a distance of \( \varepsilon \) to one another
  - Considering \( [X \cup Y]_\varepsilon \), the reunion of the clusters of two videos, is a cluster from this set contains the frames of both videos, then they are visually similar
11.1 Video Similarity

- The **Ideal Video Similarity** is the percentage of clusters in \([X \cup Y]_3\), which contain frames from both videos (relative to the total number of clusters)

\[
ivs(X, Y; \epsilon) = \frac{\sum_{x \in X} 1_{\epsilon x} \cdot 1_{\epsilon y}}{|X \cup Y|_3}
\]

11.1 IVS Calculation

- Naive calculation requires distance calculations between \(|X| \cdot |Y|\) frame pairs
- More efficient methods estimate the ivs by sampling
  - Represent each video through \(m\) randomly selected video frames
  - Estimate the ivs by the number of similar pairs \(W_{mn}\) in the samples

11.2 Voronoi Diagrams

- Voronoi diagrams (also known as Voronoi tilings) are decomposition of a metric space into disjoint parts
  - **Given:**
    - A metric space \((F, d)\)
    - A set of discrete points \(X \subseteq F\)
  - **Goal:**
    - A division of \(F\) in exactly \(|X|\) disjoint (but related with each other) parts
    - In each of these parts there is just one point from \(X\)

- **Voronoi solution:**
  - Given: a point \(z \in F\). To which part of space does \(z\) belong to?
  - Determine the point \(x \in X\), which is the closest to \(z\)
  - Then \(z\) maps the space part, where \(x\) is found
11.2 Voronoi Diagrams

- In Euclidean spaces: the set of equidistant points for each pair of points, is a **hyperplane**
  - Between each two points from X there is a hyperplane
  - Points on the left side of the hyperplane are closer to the left point while points on the right side of the hyperplane are closer to the right point

- In Euclidean spaces: the set of equidistant points of X from X there is a hyperplane

- **Voronoi Video Similarity**
  - **Voronoi diagrams** are specific geometrical layouts of spaces
  - For videos we divide the feature space according to the cluster
    - Given a video frame with \( n \) frames
      \[ X = \{ x_t : t = 1, \ldots, n \} \]
    - The Voronoi diagram \( V(X) \) of \( X \) is a division of the feature space \( F \) in \( n \) Voronoi cells \( V_X(x_t) \)

- **Voronoi Video Similarity**
  - The Voronoi cell \( V_X(x_t) \) contains all vectors in \( F \), which lie closer to the frame \( x_t \) as to all other frames of \( X \)
    \[ V_X(x_t) = \{ s \in F : g_X(s) = x_t \text{ and } x_t \in X \} \]
    with \( g_X(s) \) as the next frame from \( X \) to \( s \)
  - In the case of equal intervals of several frames one takes for \( g_X(s) \) usually the frame that is next to a predetermined point (e.g., the origin)

- Voronoi cells are combined for frames of identical clusters, therefore for \( C \in [X] \)
  \[ V_X(C) = \bigcup_{x \in C} V_X(x) \]
  is valid
11.2 Voronoi Video Similarity

- We can define similar Voronoi regions for two videos X and Y and their two Voronoi diagrams through

\[ R(X, Y; \varepsilon) = \bigcup_{d(x,y) \leq \varepsilon} V_X(x) \cap V_Y(y) \]

- If x and y are close to one another, then also their Voronoi cells will intersect. The more similar pairs there are, the greater the surface area of the \( R(X, Y; \varepsilon) \).

Example: two videos, each with two frames and their corresponding Voronoi cells. The gray area is the common area \( R(X, Y; \varepsilon) \).

The volume of \( R(X, Y; \varepsilon) \) is a measure of video similarity.

- Technical problems:
  - The Voronoi cells must be measurable (volume as a Lebesgue integral).
  - The feature space is considered compact (therefore, restricted and closed) so volumes are finite.
  - For normalization: \( \text{Vol}(F) = 1 \).

Since each cluster and Voronoi cells do not overlap, is the Voronoi video similarity:

\[ \text{vvs}(X, Y; \varepsilon) = \text{Vol}(R(X, Y; \varepsilon)) = \text{Vol}(\bigcup_{d(x,y) \leq \varepsilon} V_X(x) \cap V_Y(y)) = \sum_{d(x,y) \leq \varepsilon} \text{Vol}(V_X(x) \cap V_Y(y)) \]

Example:

- \( \text{vvs} \) in the example is 0.33, which is also consistent with the \( \text{ivs} \) in this example.
- The reason for the very good correlation is a the similar volume of each Voronoi cell.
- This correlation, is not however, generally provided.

An estimate of \( \text{vvs}(X, Y; \varepsilon) \) is possible through random sampling:

- Generate \( m \) vectors \( s_1, \ldots, s_m \) (seed vectors) independent and uniformly distributed over the space F.
- Check for each seed \( s_i \) if it is located inside \( R(X, Y; \varepsilon) \), i.e., in any Voronoi cell \( V_X(x) \) and \( V_Y(y) \) with \( d(x,y) \leq \varepsilon \).
- Let \( g_i(s_i) \) be the frame from \( X \) with the smallest distance to \( s_i \).
- Then:
  \[ s_i \in R(X, Y; \varepsilon) \iff d(g_i(s_i), g_i(s_i)) \leq \varepsilon \]
11.2 Estimation of VVS

- It is possible to describe each video \( X \), through the \( m \) tuple \( X := (g_X(s_1), ..., g_X(s_m)) \)
- \( X \) is called video signature with respect to \( S \)
- As a similarity measure for videos \( X \) and \( Y \) we can now use the degree of overlap between \( X \) and \( Y \):

\[
\text{vss}(X; Y; S, m) = \frac{\sum_{i=1}^{m} \mathbb{1}_{\{R(s_i, X, S, Y, m)\cap R(s_i, Y, S, X, m)\neq \emptyset\}}}{m}
\]

• \( \text{vss} \) is basic video signature similarity
- Since the seed vectors are uniformly distributed, the probability of event \( \text{"s} \in R(X, Y, \varepsilon)\text{"} \) represents the volume of \( R(X, Y, \varepsilon) \), thus \( \text{vvs}(X, Y, \varepsilon) \)
- \( \text{vss} \) is an unbiased estimator for \( \text{vvs} \)
- For video collections identical seeds must be used for all signature calculations

11.2 Estimation of VVS

- The number \( m \) of seeds is the signature length
  - The larger \( m \), the more accurate the estimate
  - The smaller \( m \), the easier the signature calculation
- Important issue for the election of \( m \): how high is the error probability?
  - Video database \( \Lambda \) with \( n \) and \( m \) videos seeds
  - Constant \( \gamma > 0 \) (maximum deviation)
  - \( P_{err}(m) = P(\text{the database contains at least a couple of videos, for which the difference between vvs and vss is greater than } \gamma) \)

11.2 Estimation of VVS

- Define \( \hat{\rho}(X, Y) = \text{vss}(X, Y; \varepsilon) \)
- Using Hoeffding’s inequality we can determine the maximum probability, that a sum of independent random and limited variables deviates with more than a given constant from its expected value:

\[
\text{Prob}(|\hat{\rho}(X, Y) - \rho(X, Y)| > \gamma) \leq 2 \exp\left(\frac{-2\gamma^2 m}{n}\right)
\]

- Therefore:

\[
P_{err}(m) = \text{Prob}\left( \bigcup_{X,Y \in \Lambda} |\rho(X,Y) - \hat{\rho}(X,Y)| > \gamma \right)
\]

\[
\leq \sum_{X,Y \in \Lambda} \text{Prob}(|\rho(X,Y) - \hat{\rho}(X,Y)| > \gamma)
\]

\[
\leq \frac{n}{2} \cdot 2 \exp\left(\frac{-2\gamma^2 m}{n}\right)
\]

- Sufficient conditions for \( P_{err}(m) \leq \delta \):

\[
\frac{n}{2} \cdot 2 \exp\left(\frac{-2\gamma^2 m}{n}\right) < \delta
\]

\[
m \geq \frac{2 \ln n - \ln \delta}{2\gamma^2}
\]
11.2 Estimation of VVS

\[ m \geq \frac{2 \ln n - \ln \delta}{2 \gamma^2} \]

- The bound for \( m \) is logarithmic of the size \( n \) of the video database
- The smaller the error \( \gamma \) is, the greater the values chosen for \( m \) should be

11.2 Seed Vector Generation

- The vvs is not always the same as ideal video similarity (ivs)
- ivs and vvs are the same, if the clusters are evenly distributed over the entire feature space

11.2 Seed Vector Generation

- Consider cases with ivs = 1 / 3, but too small or too high Voronoi video similarity:

11.2 Seed Vector Generation

- Goal: estimation of the ivs through basic video signatures (vss) even if ivs and vvs differ
  - Since the seeds are spread evenly throughout the feature space, the estimation is influenced by various sizes of Voronoi cells
  - Solution: distribute the seeds evenly over the Voronoi cells, regardless of their volumes

11.2 Seed Vector Generation

- To generate the seeds (rather than using the uniform distribution over \( F \)) use a distribution with density function as follows:
  - Given: two videos \( X, Y \)
  - Distribution density at \( u \in F \):
    \[ f(u; X \cup Y) = \frac{1}{\|X \cup Y\|_k} \cdot \frac{1}{\text{Vol}(V_{X \cup Y}(C))} \]
  - \( C \) denotes the cluster in \( [X \cup Y]_k \) with \( u \in V_{X \cup Y}(C) \)

11.2 Seed Vector Generation

- \( f(u; XUY) \) is inversely proportional to the volume of each cell
  - Uniform distribution on the set of clusters
- \( f(u; XUY) \) is constant within the Voronoi cell of each cluster
  - Equal distribution within each cluster
- Possible generation method for seeds:
  - Randomly choose a cluster (uniformly distributed)
  - Choose a random point within this cluster (uniformly distributed)
11.2 Seed Vector Generation

- If we do not uniformly produce seeds, but with density $f(u; X \cup Y)$, we obtain the following estimator for ivs:
  $$\sum_{d(x,y) \leq \varepsilon} \int_{X \cup Y} f(u; X \cup Y) \, du$$
- For $f(u; X \cup Y) = 1$ (uniform distribution on $F$) it is exactly the definition of $v_{vss}(X, Y; \varepsilon)$

11.2 VSS_b and IVS

- $v_{s_b}$ approximates ivs if the clusters are either identical or very good separated
- **Theorem**: Let $X$ and $Y$ be videos, so that for all pairs of clusters $c_X \in [X]_\varepsilon$ and $c_Y \in [Y]_\varepsilon$:
  - Either $c_X = c_Y$
  - Or all the frames in $c_X$ further away with more than $\varepsilon$ from all frames in $c_Y$
- Then:
  $$ivs(X, Y; \varepsilon) = \sum_{d(x,y) \leq \varepsilon} \int_{X \cup Y} f(u; X \cup Y) \, du$$

11.2 VSS_b and IVS

- Proof:
  - For each term in the sum if $d(x, y) \leq \varepsilon$, then $x$ and $y$ belong to the same cluster $C$ in $[X]_\varepsilon$ and $[Y]_\varepsilon$.
    Thus, one can rewrite the sum as follows:
    $$\sum_{d(x,y) \leq \varepsilon} \int_{X \cup Y} f(u; X \cup Y) \, du = \sum_{C \in [X]_\varepsilon \cap [Y]_\varepsilon} \sum_{u \in C} \int_{X \cup Y} f(u; X \cup Y) \, du$$
- Due to the definition of Voronoi cells, for all $z \in C$
  - with $C \in [X]_\varepsilon \cap [Y]_\varepsilon$:
    $$V_X(z) \cap V_Y(z) = V_{X \cup Y}(z)$$
  - It results in:
    $$\sum_{d(x,y) \leq \varepsilon} \int_{V_X(z) \cap V_Y(z)} f(u; X \cup Y) \, du = \sum_{C \in [X]_\varepsilon \cap [Y]_\varepsilon} \int_{V_{X \cup Y}(C)} f(u; X \cup Y) \, du$$

11.2 VSS_b and IVS

- It is not possible to use the density function $f$ for the estimation of ivs for the calculation of video signatures
  - The density function is specific for each pair of videos, but for comparison within collections, same seeds must be used
  - For this reason we use a (representative!) training set $T$ for the definition of the density function
11.2 Application

- **Algorithm** for generating a single seed:
  (m independent repetitions of the algorithm provide m seeds)
  - Given:
    - A value $\epsilon_{SV}$
    - A training set of T frames, which reflects the collection as well as possible
  - Identify all clusters $[T]_{\epsilon_{SV}}$ of set T
  - Choose any cluster $C \in [T]_{\epsilon_{SV}}$

- **Experiment:**
  - 15 videos from the “MPEG-7 content set”
    - Average length: 30 minutes
    - By means of random deletion of frames, 4 new videos were produced from each video, each having $ivs = 0.8, 0.6, 0.4$ and 0.2 when compared to the full video
  - Then the ivs was estimated through the $vss_b$
    - Two methods for generating the seeds ($m = 100$):
      (1) uniformly distributed on $F$ and
      (2) based on a test collection of 4,000 photographs from the Corel photo collection

11.2 Voronoi Gap

- $vvs$ and $ivs$ are the same, if clusters are either identical or clearly separated
  - The feature vectors are only an approximation of the visual perception, therefore, they may contain small discrepancies within visually similar clusters

- **Consider a feature space with $ivs = 1$:**
  - The Voronoi regions differ slightly, and therefore do not fill the entire feature space
11.2 Voronoi Gap

- Consider seed $s$ between the Voronoi cells
- Observation:
  - The next signature frames $g_X(s)$ and $g_Y(s)$ for two videos $X$ and $Y$ are far apart from one another: $d(g_X(s), g_Y(s)) > \varepsilon$
  - Both signature frames are similar to frames of the other videos, therefore there is an $x \in X$ with $d(x, g_Y(s)) \leq \varepsilon$ and there is an $y \in Y$ with $d(y, g_X(s)) \leq \varepsilon$

11.2 Seed Generation

- One can analytically show that for simple feature spaces the volume of the Voronoi gap can’t be neglected:
  - There are usually seeds that fall into the Voronoi gap and distort the estimate of the ivs
  - The smaller the $\varepsilon$, the smaller the Voronoi gap
  - Goal: avoid the use of seeds which (probably) lie in the Voronoi gap

- The pure definition of the Voronoi gap does not help in the verification
  - Requires distance calculations between each signature vector, and all frames of the other videos
  - Thus the efficient description of the video would be invalidated by his signature
  - It’s enough to assign probabilities for the fact that a seed is in the Voronoi gap

- Observations:
  - Both video sequences have a roughly equidistant pair of frames with respect to $s$: $(x, g_X(s))$ and $(y, g_Y(s))$
  - It is clear that the pairs themselves are dissimilar: $(x, g_X(s)) \geq \varepsilon$ and $(y, g_Y(s)) \geq \varepsilon$
  - Since the seeds in the Voronoi gap are near the borders of different Voronoi cells, one can easily find such equidistant pairs
11.2 Criteria

- Given: two videos $X, Y$ with $\mathcal{E}$-compact clusters $[X \cup Y]_\mathcal{E}$
- For every seed $s$ in the Voronoi gap, there is a vector $x \in X (y \in Y)$ with
  - $x$ is dissimilar to $g_X(s)$, therefore $d(x, g_X(s)) > \mathcal{E}$
  - $x$ and $g_X(s)$ are equidistant from $s$, particularly $d(x, s) - d(g_X(s), s) \leq 2\mathcal{E}$

11.2 Criteria

- Proof:
  - Since $s$ is in the Voronoi gap, we have $d(g_X(s), g_Y(s)) > \mathcal{E}$
  - Since clusters are by assumption $\mathcal{E}$-compact, $g_X(s)$ can’t be in the same cluster as $x$ and $g_Y(s)$, therefore $d(g_Y(s), x) > \mathcal{E}$
  
  - Further: $d(x, s) - d(g_X(s), s) \leq d(x, g_Y(s)) + d(g_Y(s), s) - d(g_X(s), s) \leq \mathcal{E} + d(g_Y(s), s) - d(g_X(s), s) + 2\mathcal{E}$

11.2 Application

- Define a ranking function $Q$ for the signature vector:
  $$Q(g_X(s)) - \min_{x \in X, d(x, g_X(s)) > \mathcal{E}} d(x, s) - d(g_X(s), s)$$
- The further away are seeds from the borders of Voronoi cells, the higher the value of $Q(g_X(s))$

- Test whether a seed $s$ is in the Voronoi gap between a video $X$ and any other random sequence:
  - If there is no vector $x \in X$ with
    - $x$ is dissimilar to $g_X(s)$ and
    - $d(x, s) - d(g_X(s), s) \leq 2\mathcal{E}$,
  then $s$ is never in the Voronoi gap between $X$ and another video
11.2 Application

- "Safe" seeds have Q-values $> 2\epsilon$
- This is not required but sufficient, and often difficult to find
  - In general, many seeds with Q-value $\leq 2\epsilon$ are not in the Voronoi gap
- Generate various seeds and choose only the ones with the best Q-values

- The symmetrical $\text{vss}_s$ between two videos is defined by the seeds with the highest ranking in $X_s$ and $Y_s$
  \[ \text{vss}_s(X_s, Y_s; m) = \frac{1}{|m|} \sum_{i=1}^{m} \frac{1}{|m|} \sum_{j=1}^{m} 1 - \text{dist}(x_i, y_j) \]
  - With $j[1], \ldots, j[m']$ and $k[1], \ldots, k[m']$ as the rankings of the signature frame in the $X_s$ and $Y_s$
    (e.g., $Q(x_{s,j[1]}) \geq \ldots \geq Q(x_{s,j[m']})$)

- The asymmetric form leads to some distortion in the estimate
  - If a video is a partial sequence of another video, the asymmetric $\text{vss}_s$ is significantly higher when calculated with the shorter video, rather than with the longer one
  - Allows more efficient implementations

- Database of short video clips from the Web
- Compared with manual evaluation

11.2 Application

- Let $m' > m$ the number of frames in the video signature
  - Generate $X_s$ with a set of $m'$ seed vectors
  - Then compute $Q(g(x)(s))$ for all $g(x)(s)$ from $X_s$ and arrange the $g(x)(s)$ according to decreasing Q-value
- Analogous to $\text{vss}_s$, we can now define ranked video similarity $\text{vss}_r$

- $\text{vss}_r$ uses 50% of the frames with the highest ranking in $X_s$ for comparison with the corresponding frames in $Y_s$ and 50% of the frames with the highest ranking in $Y_s$ for comparison with the corresponding frames in the $X_s$
  - Overall, again only $m$ comparisons
  - Alternatively we can also use an asymmetric $\text{vss}_r$ with $m$ seeds with the highest ranking with respect to just one video

11.2 Retrieval Effectivity: $\text{VSS}_B$ vs. $\text{VSS}_R$

- Database of short video clips from the Web
- Compared with manual evaluation
Next lecture

- Video Abstraction
  - Video Skimming
  - Video Highlighting
  - Skimming vs. Highlighting