11 Video Similarity

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)

11.1.1 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
11.1 Video Similarity

- 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”
  - (Wu, Ngu and Hauptmann, 2006) define ‘near duplicates’
- 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 Video Similarity

- Magnitude of the problem: video redundancy on the web

11.1 Video Signatures

- Idea: select a small number of features that represent a video with minimal errors
  - Minimize the distance between the video and its representation
  - Example:
    - Features as vectors in \( \mathbb{R}^n \)
    - Euclidean distance
    - Method of least squares (k-means)
    - Best cluster representatives (k-medoids)

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

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

– 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), \( \text{nvs} = 1 \)
– \( \text{nvs} = 0 \), 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
  – E.g., generate \( Y \) through multiplication of a single frame from \( X \). For \( |Y| >> |X| \) \( \text{nvs}(X, Y, \varepsilon) \approx 1 \)

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 \( d(x, z) \)?
11.1 Video Similarity

- 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 union 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 \( [X \cup Y]_\varepsilon \), which contain frames from both videos (relative to the total number of clusters).

\[
ivs(X, Y; \varepsilon) = \frac{\sum_{x \in X} \sum_{y \in Y} \mathbb{1}_{d(x, y) < \varepsilon}}{|[X \cup Y]_\varepsilon|}
\]

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 \( W_m \) in the samples.

11.1 IVS Calculation

- Small values of \( m \) speed up calculation, but may distort the results.
  - Consider two videos \( X \) and \( Y \) are of the same length.
  - For each frame in \( X \) there is exactly one similar frame in \( Y \) (and vice-versa).
    - Therefore \( ivs = 1 \).
  - The expected value of the number of similar pairs in a sample of size \( m \) is \( E(W_m) = m^2/|X| \).
  - Thus it takes an average of \( \sqrt{|X|} \) samples to find on average at least a similar pair.

- Other solutions? Voronoi diagrams.

11.2 Voronoi

- Georgi Voronoi: Russian mathematician
  - Known for the Voronoi diagrams: decomposition of a metric space into disjoint parts.
  - Starting from a:
    - \( \cdots \) metric space \( (F, d) \)
    - \( \cdots \) set of discrete points \( X \subseteq F \)
  - Goal:
    - Divide \( F \) in exactly \( |X| \) disjoint parts.
    - In each of these parts there is just one point from \( X \).

\[\text{Georgi Voronoi}\]
11.2 Voronoi Diagrams

- Voronoi’s tessellation:
  - Each point in the $x_i \in X$ region is closer to $x_i$ than to any other $x_j \in X$ with $j \neq i$
  - 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 $z$
- In Euclidean spaces: the set of equidistant points for each pair of points, forms a hyperplane

11.2 Voronoi Video Similarity

- Applications such as in the analysis of the growth of crystals
- Simple algorithmic calculation ($n^2$) of Voronoi diagrams by grouping areas
  - For a fixed point calculate all the dividing hyperplanes; Merging the planes results in the Voronoi cell
  - More efficient algorithms exist e.g., in the Euclidean case: running time $O(n \log n)$

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

- 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 closest 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 $C \subseteq [X]$, $V_X(C) = \bigcup_{x_t \in C} V_X(x_t)$ 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
  \[ R(X, Y; \epsilon) = \bigcup_{d(x, y) \leq \epsilon} V_X(x) \cap V_Y(y) \]
  - 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 \( R(X, Y; \epsilon) \).

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

- The volume of \( R(X, Y; \epsilon) \) 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 both the clusters and the Voronoi cells don't overlap, the Voronoi video similarity is:
  \[ \text{vvs}(X, Y; \epsilon) = \text{Vol}(R(X, Y; \epsilon)) = \sum \text{Vol}(V_X(x) \cap V_Y(y)) \]

11.2 Example

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

11.2 Estimation of VVS

- An estimate of \( \text{vvs}(X, Y; \epsilon) \) 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 \( R(X, Y; \epsilon) \), i.e., in any Voronoi cell \( V_X(x) \) and \( V_Y(y) \) with \( d(x, y) \leq \epsilon \).
  - Let \( g_X(s_i) \) be the frame from X with the smallest distance to \( s_i \).
  - Then: \( s_i \in R(X, Y; \epsilon) \) iff \( d(g_X(s_i), g_Y(s_i)) \leq \epsilon \).
It is possible to describe each video \( X \), through the tuple \( \mathbf{X}_s := (g_X(s_1), \ldots, g_X(s_m)) \). 

\( \mathbf{X}_s \) 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 \( \mathbf{X}_s \) and \( \mathbf{Y}_s \):

\[
\text{vss}_b(X, Y; \epsilon, m) = \frac{\sum_{i=1}^{m} 1(\epsilon(g_X(s_i), g_Y(s_i)) \leq \epsilon)}{m}
\]

\( \text{vss}_b \): basic video signature similarity

Since the seed vectors are uniformly distributed, the probability of event “\( s \in R(X, Y, \epsilon) \)” represents the volume of \( R(X, Y, \epsilon) \), thus \( \text{vss}(X, Y, \epsilon) \).

\( \text{vss}_b \) is an unbiased estimator for \( \text{vss} \).

For video collections identical seeds must be used for all signature calculations.

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 selection of \( m \): how high is the error probability?

– Video database \( \Lambda \) with \( n \) videos and \( m \) seeds
– Constant \( \gamma > 0 \) (maximum deviation)
– \( P_{\text{err}}(m) = P \left( \text{the database contains at least a couple of videos, for which the difference between \( \text{vss} \) and \( \text{vss}_b \) is greater than \( \gamma \) } \right) \)

A sufficient condition to guarantee for \( P_{\text{err}}(m) \leq \delta \) is the choice of \( m \) as

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

Proof: next slides

Define \( \rho(X, Y) = \text{vss}(X, Y; \epsilon) \)

\( \hat{\rho}(X, Y) = \text{vss}_b(X, Y; \epsilon, m) \)

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:

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

Therefore:

\[
P_{\text{err}}(m) \leq \frac{\gamma}{m} \leq \frac{2 \ln n - \ln \delta}{2 \gamma^2m}
\]

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

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

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.

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

### 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 \( u \in F \):
    \[
f(u; X \cup Y) = \frac{1}{||X \cup Y||_e} \cdot \frac{1}{\text{Vol}(V_{X \cup Y}(C))}
\]
  - \( C \) denotes the cluster in \([X \cup Y]_e\) 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 produced seeds, but with density \( f(u; X \cup Y) \), we obtain the following estimator for \( ivs \):
  \[
  \sum_{d(x,y) \leq \varepsilon} \int_{V_X(x) \cap V_Y(y)} f(u; X \cup Y) \, du
  \]
  - For \( f(u; X \cup Y) = 1 \) (uniform distribution on \( F \)) it is exactly the definition of \( vvs(X, Y; \varepsilon) \)

11.2 VSS\(_B\) and IVS

- \( vss \) 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:
  \[
  ivs(X, Y; \varepsilon) = \sum_{d(x,y) \leq \varepsilon} \int_{V_X(x) \cap V_Y(y)} f(u; X \cup Y) \, du
  \]

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:
    \[
    \sum_{d(x,y) \leq \varepsilon} \int_{V_X(x) \cap V_Y(y)} f(u; X \cup Y) \, du
    \]
    \[
    = \sum_{C \in [X]_{\varepsilon} \cap [Y]_{\varepsilon}} \int_{V_X(x) \cap V_Y(y)} f(u; X \cup Y) \, du
    \]

11.2 VSS\(_B\) and IVS

- Since \( [X]_{\varepsilon} \cap [Y]_{\varepsilon} \) is the set of similar clusters in \( [X \cup Y]_{\varepsilon} \), the last term is just the \( ivs \)

11.2 Application

- 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 $\ell_{SV}$
    - A training set of $T$ frames, which reflect the collection as well as possible
  - Identify all clusters $[T]_{\ell_{SV}}$ of set $T$
  - Choose any cluster $C \in [T]_{\ell_{SV}}$

11.2 Application

- Create a seed in the Voronoi cell of the selected cluster
  - Generate random vectors over the feature space, until one of them is in $V(C)$
  - (to simplify this procedure, one can also use a random frame from $C$ as seed)

11.2 Application

- **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
    - 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

<table>
<thead>
<tr>
<th>Seed Vectors</th>
<th>Uniform Random</th>
<th>Corel Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>IVS</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>Average</td>
<td>0.573</td>
<td>0.499</td>
</tr>
<tr>
<td>StdDev</td>
<td>0.416</td>
<td>0.306</td>
</tr>
</tbody>
</table>

11.2 Voronoi Gap

- 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

- $\nu_{SV}$ 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

11.2 Voronoi Gap

- In this example: since the $\nu_{SV}$ is defined by the similar Voronoi regions, it is **strictly smaller** than ivs
  - The difference is calculated using the offset (the free space)
    - The greater the difference, the more underestimates $\nu_{SV}$ the ivs
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 \( d(y, g_X(s)) \leq \varepsilon \)

- Therefore: seeds between Voronoi cells can cause dissimilar signature vector pairs, even if both vectors have similar partners in the other videos

- The Voronoi Gap \( G(X, Y; \varepsilon) \) for videos \( X \) and \( Y \) is the set of all \( s \in F \) with:
  - \( d(g_X(s), g_Y(s)) > \varepsilon \)
  - There is an \( x \in X \) with \( d(x, g_Y(s)) \leq \varepsilon \)
  - There is an \( y \in Y \) with \( d(y, g_X(s)) \leq \varepsilon \)

11.2 Seed Generation

- If we randomly generate \( m \) seeds of which \( n \) lie in the Voronoi gap, then is \( \text{vs} \) of the remaining \( (m - n) \) vectors exactly the IvS

- Problem: how to efficiently recognize whether the vector lies in the Voronoi gap?

- 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

- Observation
  - 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 Criterion

• Given: two videos \( X, Y \) with \( \varepsilon \)-compact clusters \([X \cup Y]\) \( \varepsilon \)
• 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 \( d(x, g_x(s)) > \varepsilon \)
  – \( x \) and \( g_x(s) \) are equidistant from \( s \), particularly \( d(x, s) - d(g_x(s), s) \leq 2\varepsilon \)

11.2 Criterion

– Since \( s \) is in the Voronoi gap, there is a \( y \in Y \) with \( d(y, g_x(s)) \leq \varepsilon \), and due to the definition of \( g \)
  \( d(g_y(s), s) \leq d(y, s) \)
  – So one can estimate \( g_y(s) \) through \( y \). The triangle inequality yields:
  \[ d(x, s) - d(g_x(s), s) \leq \varepsilon + d(g_y(s), s) - d(g_x(s), s) \leq \varepsilon + d(y, g_y(s)) \]

11.2 Criterion

• 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
    * \( d(x, s) - d(g_x(s), s) \leq 2\varepsilon \),
    then \( s \) is never in the Voronoi gap between \( X \) and another video

11.2 Application

• Define a ranking function \( Q \) for the signature vector:
  \[ Q(g_x(s)) = \min_{x \in X, d(x, g_x(s)) \leq \varepsilon} d(x, s) - d(g_x(s), s) \]
  – The further away are seeds from the borders of Voronoi cells, the higher the value of \( Q(g_x(s)) \)

11.2 Application

• Higher values of \( Q \) are bright, lower values are dark
11.2 Application

• “Safe” seeds have Q-values > 2ε
• This is not required but sufficient, and often difficult to find
  – In general, many seeds with Q-value ≤ 2 ε are not in the Voronoi gap
• Generate various seeds and choose only the ones with the best Q-values

11.2 Application

• Let m’ > m be the number of frames in the video signature
  – Generate X with a set of m’ seed vectors
  – Then compute Q(g(s)) for all g(s) from X and arrange the g(s) according to decreasing Q-value
• Analogous to vss, we can now define ranked video similarity vss_r

11.2 Application

• The symmetrical vss, between two videos is defined by the seeds with the highest ranking in X and Y:
  \[ vss(X, Y; \epsilon, m) = \frac{1}{m} \sum_{i=1}^{m} \sum_{s=1}^{[m/2]} 1\{d(gX(s), gY(s)) \leq \epsilon\} + \sum_{s=1}^{[m/2]} 1\{d(gX(s), gY(s)) > \epsilon\} \]
  – With j[1], ..., j[m'] and k[1], ..., k[m'] as the rankings of the signature frame in the X and Y
  (e.g., Q(gX(s_j)) ≥ ... ≥ Q(gX(s_j)))

11.2 Application

• The asymmetric form leads to some distortion in the estimate
  – If a video is a partial sequence of another video, the asymmetric vss, is significantly higher when calculated with the shorter video, rather than with the longer one
  – Allows more efficient implementations

11.2 Application

• Database of short video clips from the Web
• Based on manual tagging

11.2 Application

• vss, uses 50% of the frames with the highest ranking in X, for comparison with the corresponding frames in Y, and 50% of the frames with the highest ranking in Y, for comparison with the corresponding frames in the X
  – Overall, again only m comparisons
  – Alternatively we can also use an asymmetric vss, with m seeds with the highest ranking with respect to just one video

11.2 Retrieval Effectivity: VSS vs. VSS

• The asymmetric form leads to some distortion in the estimate
  – If a video is a partial sequence of another video, the asymmetric vss, is significantly higher when calculated with the shorter video, rather than with the longer one
  – Allows more efficient implementations
• Video Similarity
  – The naive approach
  – Voronoi Video Similarity

Next lecture

• Video Abstraction
  – Video Skimming
  – Video Highlighting
  – Skimming vs. Highlighting