Content-Based Retrieval of Visual Media

The following are important features of current visual media search systems.

- Successful content-based retrieval systems are generally task or domain specific, i.e. they are developed for a specific activity such as spotting cars in a video or tracking football players in match.
  They can often be adapted for use in different domains, but they are not general purpose.

- Automatic understanding tools have been (to date) impossible to develop. In terms of “intelligence” systems are very basic.
  Systems are thus typically highly interactive involving users developing queries through a series of iterative searches to steer the system towards relevant items (“human-in-the-loop”).

Content-Based Retrieval of Visual Media

A finding from early work on content-based retrieval of visual media is that searchers should be given simple tasks which they can perform consistently. Rather than complex ones which can be confusing and lead to variable outputs.

e.g. they could be asked to identify relevant features of retrieved images in a simple way in order to refine the query when performing multiple search stages.

They should not be required to understand the science of image analysis or describe why an image or parts of it are relevant in a detailed and personally subjective way (which is very likely to produce inconsistent behaviour).

Issues in Visual Media Retrieval

- Multimedia objects have multiple dimensions:
  - How we view an object depends on: what our task is or what we are looking for.
- To allow different interpretations from different potential searchers, we must capture all these possible features when indexing the content.
  e.g. Is this a boat or a sunset?

Issues in Visual Media Retrieval

What features could a user search over here?

Issues in Visual Media Retrieval

- Each feature used in a visual retrieval system generates a ranked listing of items. based on a match between a query image and the image collection.
- The set of ranked lists must be integrated into a single overall ranked list combining the evidence for each individual feature into an overall similarity value.
- Labelling of features whether automatic or manual will often contain errors, or in the case of manual annotations inconsistencies.

Issues in Visual Media Retrieval

- We must also understand that query specifications will be incomplete, e.g. find a picture “like” this one. What does the user mean by “like”? We can’t tell what the user’s interest in the image is, and we don’t know which of the, often many, interpretations of an image to concentrate on.
- Visual media is very content rich with many possible interpretations, and can thus often address many diverse information needs.
- Query specifications may be refined in cycles of relevance feedback - as with text document retrieval.

The Semantic Gap

The *Semantic Gap* refers to the gap between the contents of a multimedia data stream and its meaning as interpreted by human observers.

As will be illustrated in this section, it is relatively easy to extract low level features from visual images, e.g. the colours presentation in an image,

but very much harder to automatically describe images in the way that a human observer would.

This difference between machine and human descriptions of visual media is referred to as the *semantic gap.*

The Semantic Gap

Consider the following examples of gaps in representation and interpretation:

- There is little gap between a table of salary numbers and their meaning.
- There is a larger gap between the words in a document and its overall information meaning.
- There is a much larger gap between the features of a video which can be extracted automatically (as described in these notes) and its meaning or semantic interpretation by a user.

Multimedia information retrieval for visual content is concerned with developing technologies for practical systems which support effective search, while realising that current visual analysis and interpretation technologies only enable the semantic gap to be closed in very constrained situations.

Content-Based Retrieval of Image data

Application areas for Content-Based Retrieval of Images include:

- art galleries
- architectural/engineering/interior design
- remote sensing
- geographical information systems
- weather forecasting
- fabric/fashion design
- trademark management
- law enforcement (photos, fingerprints)
- picture archiving, etc.

Content-Based Retrieval of Image data

- Relevant images can be located using a variety of attributes.
- These can include automatically extracted and manually assigned features.
- Depending on the type of feature, retrieval can be carried out on three different levels:
  - Level 1 - using image primitives based on extracted features.
  - Level 2 - based on derived attributes (iconography).
  - Level 3 - inferred abstract attributes (iconology).

Text Based Indexing

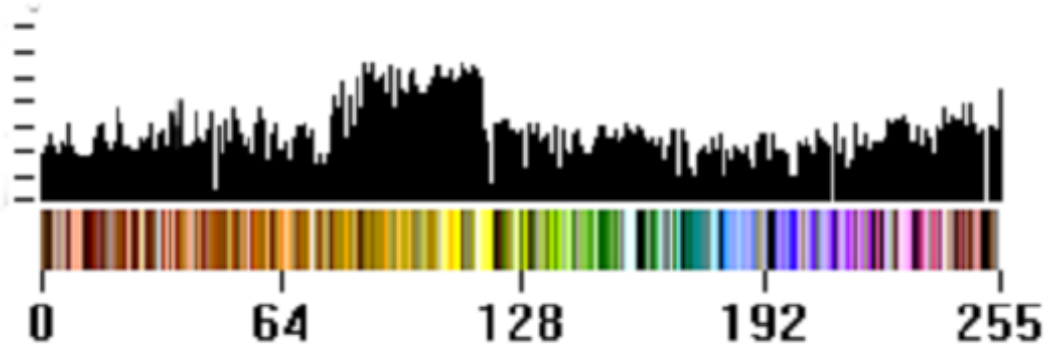
Some possible sources of information for use in the retrieval process:

- Alternate text for image on a web page
- Image URL e.g. http://www.cars.com/alfa_romeo/alfa_156.jpg will have the terms cars, alfa, romeo, and 156 extracted.
- Same paragraph text - use text near image as probably related to the image - weight nearer text higher.
- Document or image “Title”
- Heading - most recent heading in a document prior to the image.
- Anchor text - text pointing to the image.
- Other terms - Any term on the page with the image.

Level 1 - extracted image primitives

The lowest level (pre-iconography), using image primitives based on extracted features such as:

- **colour** ... the full RGB spectrum or perhaps segmented into bands, finds all images containing yellow and blue ... An image containing blue/green, yellow and orange could be a sunset!



- This is a colour histogram ... shows a profile of in the image.

Level 1 - extracted image primitives

texture - what is texture? There is no precise definition, but it is at least the following:

- An attribute representing the spatial arrangement of grey pixels (relates to texture NOT colour).
- A measure of properties such as smoothness, coarseness and regularity.
- Repetition of basic texture elements called texels which contain several pixels, whose placement could be periodic, quasi-periodic, or random.
- Natural textures are generally random whereas artificial textures are often periodic.
- Texture may be coarse, fine, smooth, granulated, rippled, regular, irregular or linear.

Level 1 - extracted image primitives

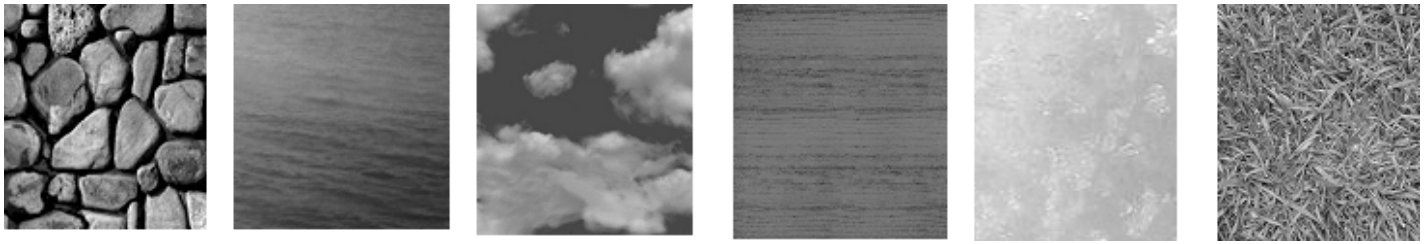
- Spatially extended patterns based on the more or less accurate repetition of some unit cell such as a texton or subpattern.

So a query could be: find images with regions of texture similar to grass.

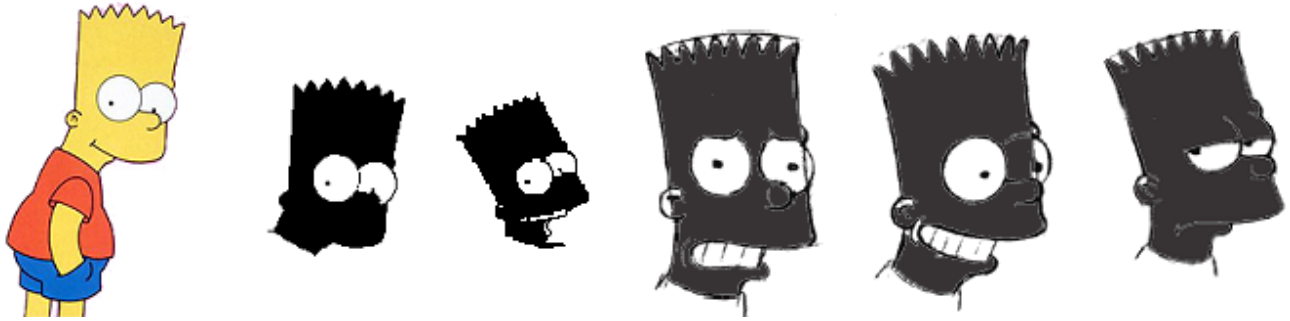


Textures & Shape

texture they are really black and white features

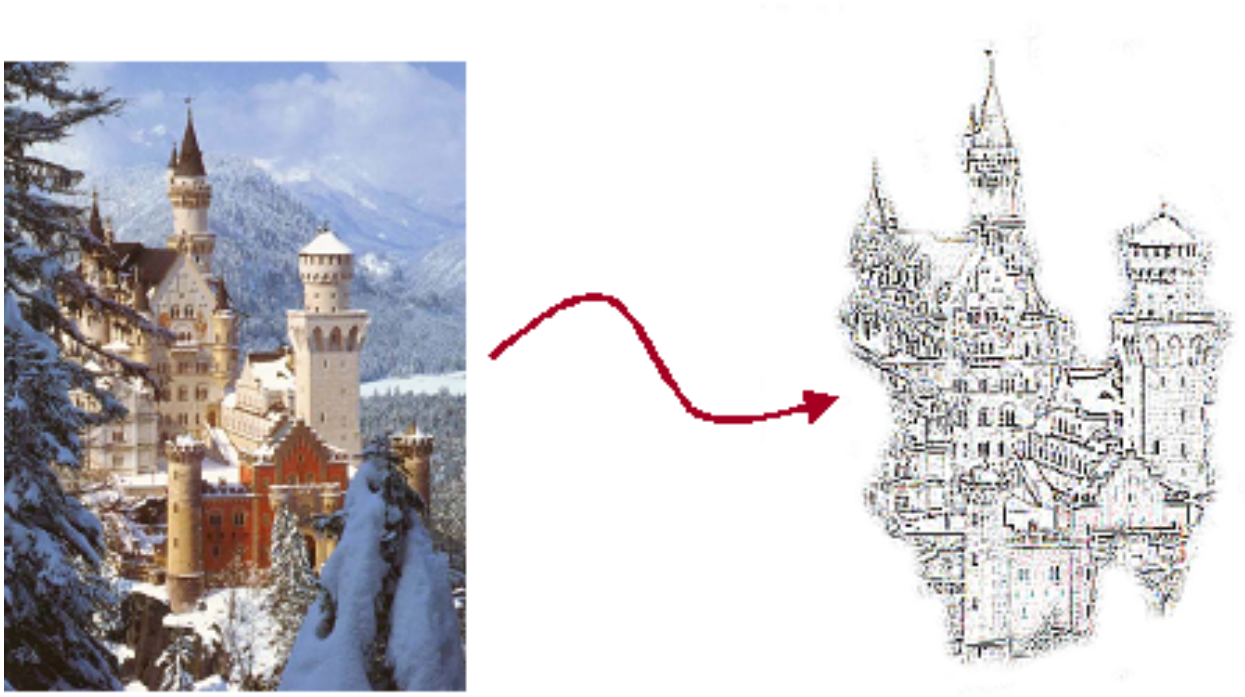


shape - geometric shapes, 2D ... find images containing a shape similar to this sketch ... example application is a Bart detector ...



Level 1 - extracted image primitives

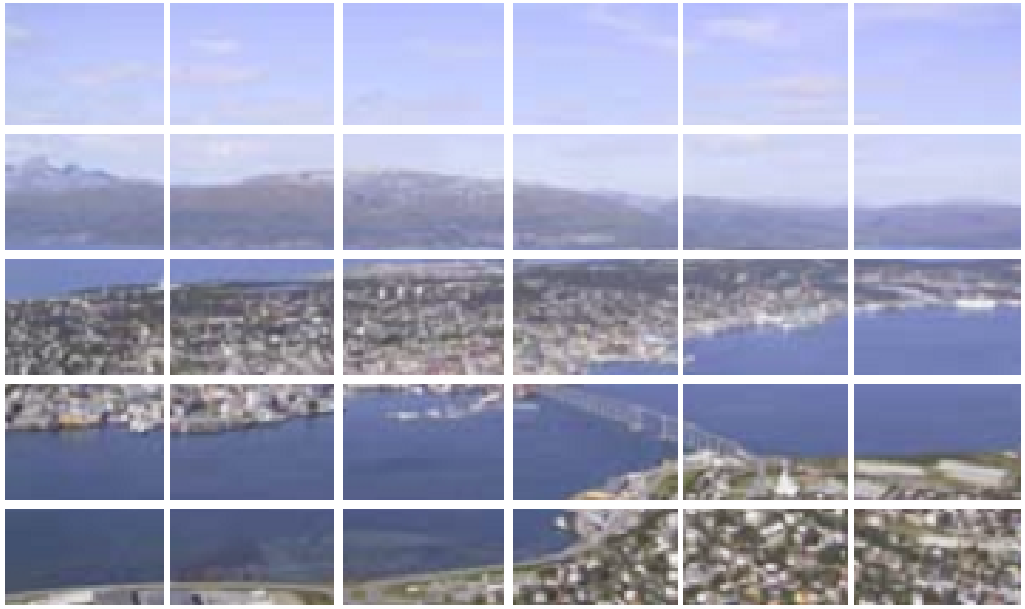
shape - complex shapes can be difficult to process



Level 1 - extracted image primitives

spatial placement - X-Y co-ordinates in a rectangle ... find images containing some object or interest in the top left corner...

or image with sky texture in top of picture and water near bottom



Level 1 - extracted image primitives

Some other primitive feature examples:

- Text within images - captions in newscast frames
- Domain concepts - noses, eye colour in faces, depressions in weather maps, etc.
- Spatial relationships - X in_front_of Y, Y behind Z, etc.
- But the one which works generally a combination of the above - e.g. find images with yellow triangles arranged in a circle.

Level 1 - extracted image primitives

Another example. Find images with sky texture and colour in the top half of the image, with a water texture in the bottom right and with an aeroplane shape in the middle of the image.

Level 2 - iconography

- In images we can also have derived attributes such as the presence of specific objects, e.g. chairs around a table or named specific individuals and this leads to level 2 type features.
- Level 2 is retrieval based on derived attributes (iconography: describing a picture's actual contents, or icons).
- This would include queries like find pictures of a train crossing a bridge or pictures of Bill Clinton meeting Gerry Adams

Level 2 - iconography

“Picture of Bill Clinton meeting Gerry Adams”

“Picture of a wine car in front of a house”

Level 3 - iconology

- More abstract still, inferred abstract attributes which do not correspond directly to content in the image, but to some inferred attribute,

  e.g. if we have football players and a goalpost and a football in a picture then we have a “football match”. This is level 3 (iconology: describing a picture’s deeper artistic significance)

- This corresponds to queries like *find images of a football match*, as opposed to *find images of 2 footballers and a football*, as opposed to *images of green with a grass texture and two splashes of black and white in a striped arrangement with a black/white circular pattern in the middle of the two splashes*.

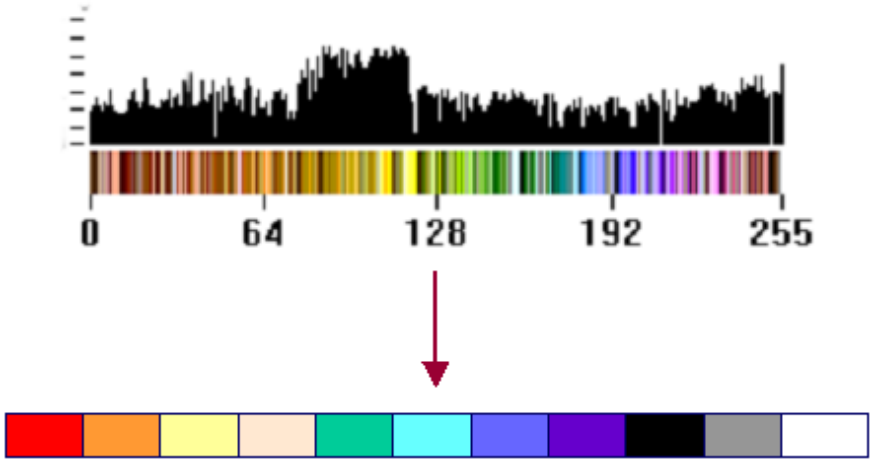
Level 3 - iconology

“Picture of a motor race”

“Picture of a storm over a city”

Using Colour Histograms

- A problem with retrieval based on colour is that small variations in colour can lead to non-matches.
- This problem is addressed by grouping perceptually similar colours together.
- Colour histogram based retrieval is the easiest and most obvious approach to doing this.



Using Colour Histograms

Example of textual markup description of colour histogram.

```
<colour>
<red> 0.8 </red>
<orange> 0.4 </orange>
<yellow> 0.3 </yellow>
<green> 0.2 </green>
<blue> 0.1 </blue>
<indigo> 0.5 </indigo>
<violet> 0.4 </violet>
</colour>
```

A separate description could be generated for each region in a divided image.

Using Colour Histograms

Similar analysis can be carried out for all images in a collection, and a query image.

The query image is then compared to each image in the collection.

There are many possible ways of computing a similarity measure between the query and each image.

One simple measure is to compute the difference between each field of the query and the each image, and then compute the sum of the differences.

Images are then ranked in increasing order to sum of difference, i.e. the image with the smallest sum of differences is the most similar to the query.

Texture-Based Retrieval

Example of textual markup description of texture.

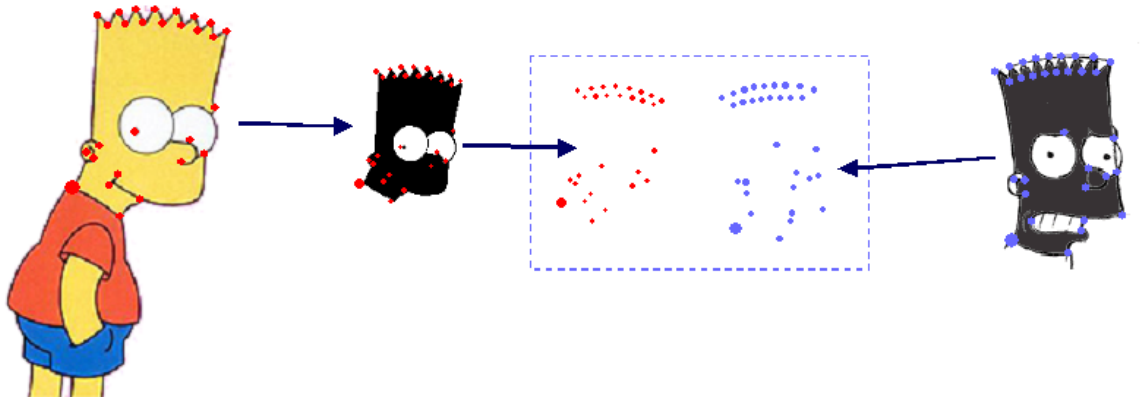
```
<texture>
<sea> 0.1 </sea>
<wood> 0.7 </wood>
<grass> 0.3 </grass>
</texture>
```

Many other texture models can be defined for comparison, e.g. sky, concrete.

Or the above one may be subdivided, e.g. into different types of wood.

Shape-Based Retrieval

- Important for meteorology, medicine, manufacturing, law, etc.
- To keep things simple look at 2D shapes only, but there are issues:
  - shapes are identified by determining boundary points and collecting these into feature descriptors / vectors.
  - similarity between shapes is measured in distance between feature vectors.
  - retrieval requires a query or sample shape as a starting point

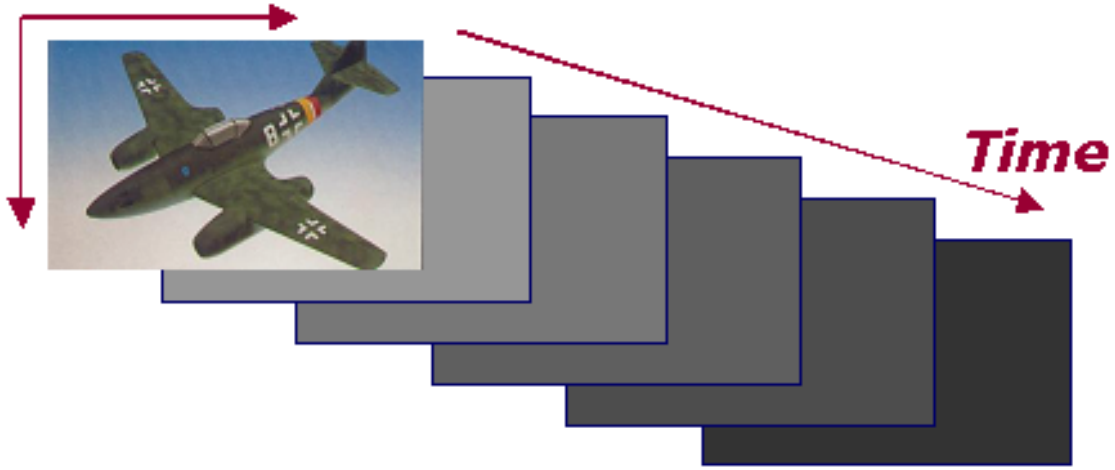


Content-Based Retrieval of Video

- Video is a continuous media.
- Events in video add a temporal dimension to those in single images.
- Video is usually combined with an audio soundtrack.
- For most effective access an application should generally use both the video and audio media synchronised for retrieval.
- Some searching on video can be achieved using metadata such as: date, title, director plus a textual description of movie contents, but this gives only very limited searching functionality.
  - much more interesting is “content-based search”.

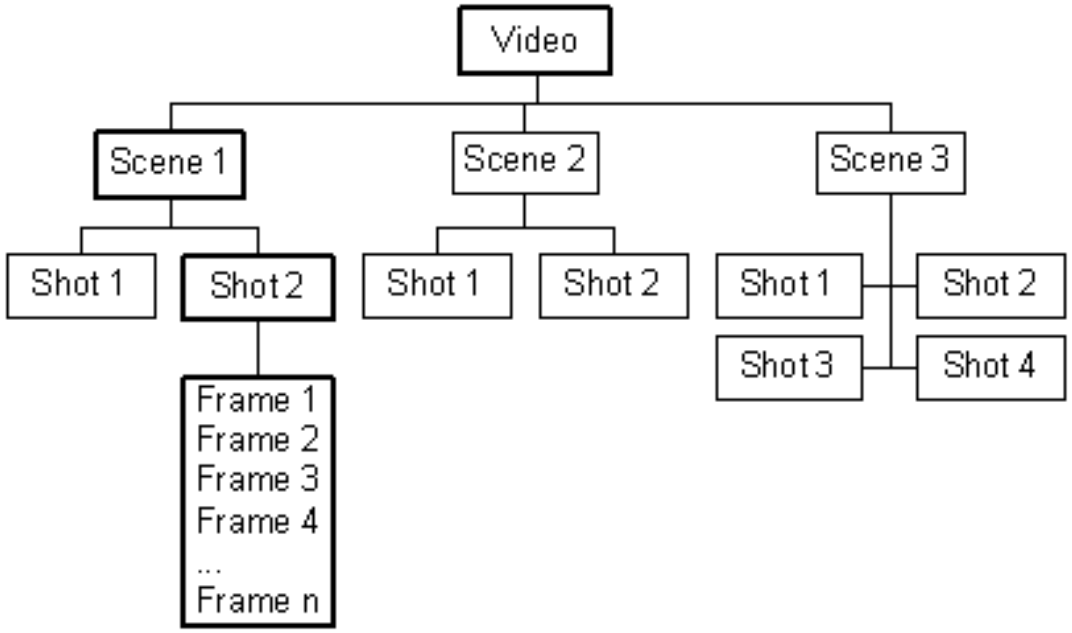
What is Video?

- Video is a sequence of individual shots combined together in some way.
- Video is of variable lengths and played as a continuous stream into a 2D window.
- Thus it has 3 dimensions: x, y and time.
- To do video retrieval we must identify the clips and then segment the video into a list of clips.



Scenes and Shots

- Video is typically composed of a series of scenes.
- Most scenes can be decomposed into a sequence of shots.
- A shot in video information is a sequence of continuous images (frames) from a single camera.

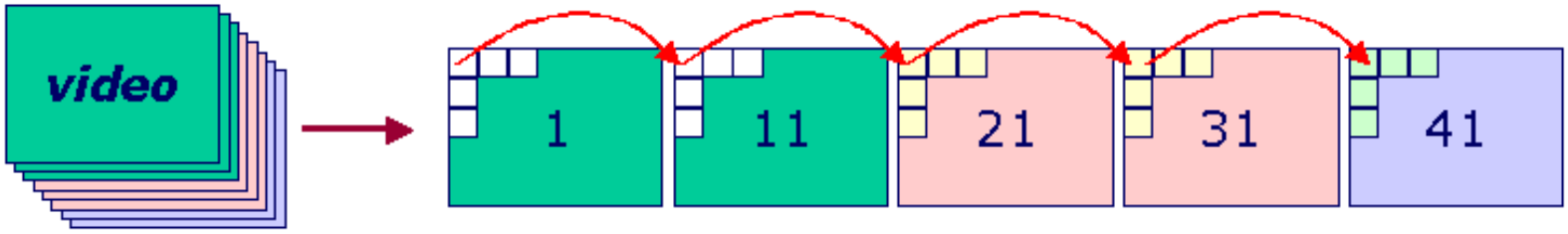


Shot Boundary Detection

- A shot boundary is crossed when a new camera is used, or a recording instance ends and a new one begins.
- Shot Boundary Detection automatically segments video into its constituent shots.
- Why do this?
  - Allows content-based operations over video at granularity of shot units, e.g. browsing, searching.

Shot Boundary Detection

- How? By examining every X frames / adjacent frames to look for shot cuts.
  - Simple shot cuts are easy to process - Based on colour, texture, intensity/brightness, etc



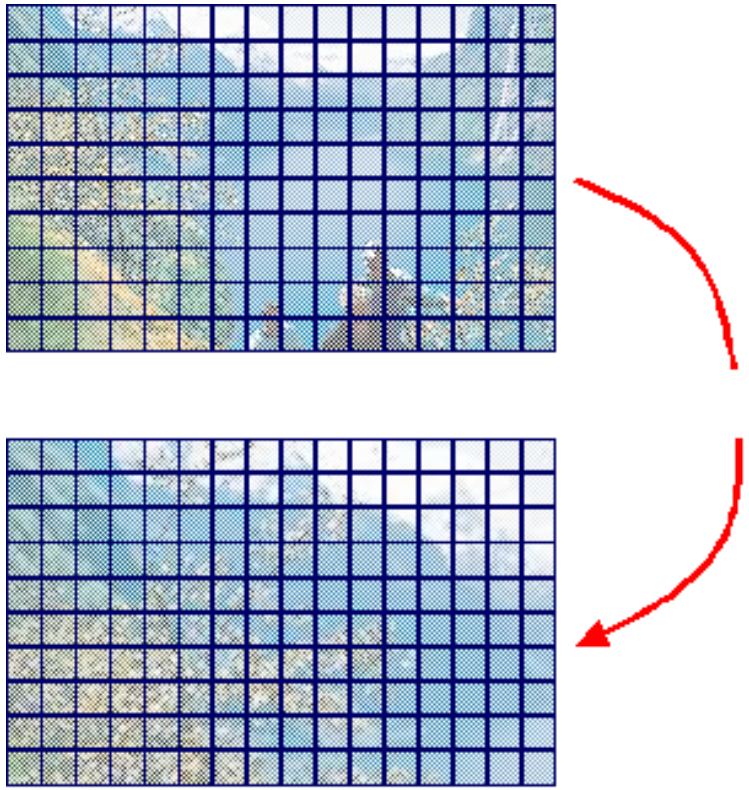
Shot Boundary Detection

But it can be quite difficult because of camera tricks:

- Dropped Shot Boundaries:
  - fade-in and fade-out
  - dissolving
  - morphing
  - wipes
  - many other chromatic effects
- False Shot Boundaries:
  - zooming and panning
  - tilting
  - booming and tracking
  - events in the content itself - camera flashes

Shot Boundary Detection

Example of “zooming”.



Browsing Digital Video

How should we display video for browsing?

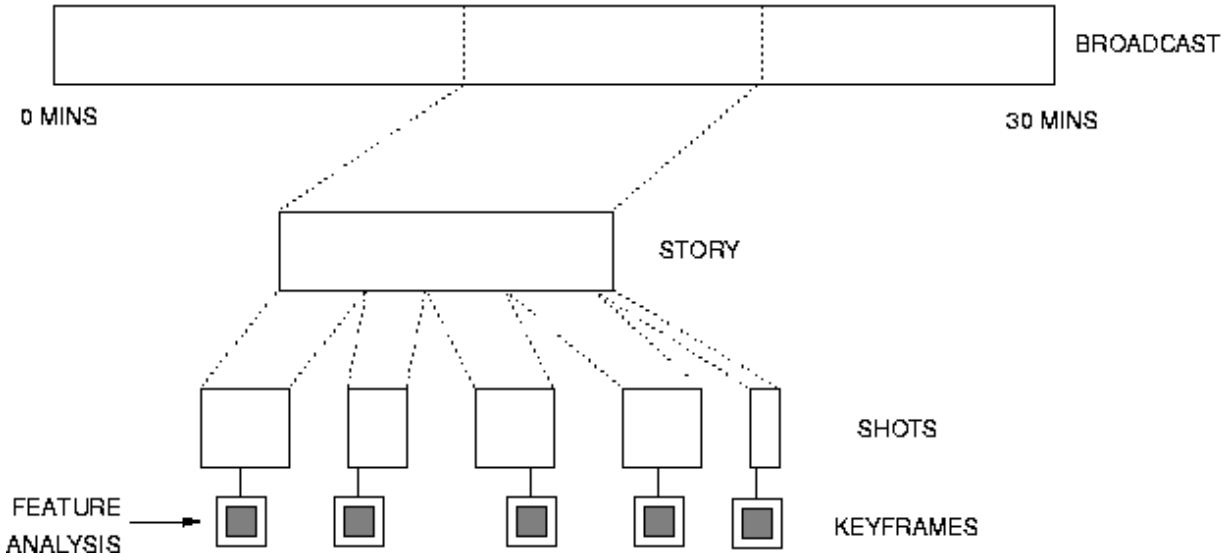
- DVDs allow random access:
  - Based on chapters.
  - Which have been segmented manually - chapters are represented by keyframes or video clips.
- A problem is that browsing keyframes is still browsing through a lot of video (hours)
  - but shot-level granularity and keyframe browsing is relatively easy to achieve.

Keyframes

How do you identify a representative keyframe for a shot?

- Random frame is hit and miss.
- First or Last frame is also problematic.
- Choosing the middle frame from a shot is the easiest approach.
- Choosing the frame with the most average colour histogram from the shot is the way Fschlar does it.
- A virtual centroid - a frame that does not actually exist, but contains the average of all colour data in the shot.
- So now we can represent a video as a sequence of keyframes, one for each shot.

Structuring Broadcast News Content



Breakdown of a news broadcast into stories, shots and keyframes.

- Shots are defined by scene or camera changes.
- Ideally the keyframe is the single frame which “best” represents the visual information in the shot.
- Shots are indexed by analysing the visual features of the keyframe.

Multimedia Indexing

- Spoken content:
  - convert into text using speech recognition (or if available use close-captions subtitles) and then apply text based information retrieval.
- Extract features from representative (?) keyframes from each shot:
  - Low-level features: colours, textures, edges.
  - Determine the presence of people, faces.
  - Classify images as: indoor, outdoor, landscape, cityscape.
  - Idenify more difficult features (presence of objects, landscapes or people tracking).
  - Classify audio into: spoken data, music, silence.

Multimedia Indexing

- Keyframes can be examined to extract useful additional features.
- Depending on the domain of the data various useful features can be identified.
- For many applications a very useful feature is face detection.
- For structured content such as news data. Text appearing in the video images can be extracted using “video OCR”, and then added to the document text index data. This is effectively an additional source of metadata for the images.

The following examples are taken from the *Informedia* project at Carnegie Mellon University.

Informedia: Face and Text Detection

Informedia: Face and Text Detection

Detects faces and text in video keyframes.

- Requires models of typical faces, for example to find standard features such as eyes, nose and mouth.
- Ability to detect text in the frame and segment the text region.

In the example we can see examples of success and failure in object detection.

Failure to correctly identify a face or a piece of text typically results from difficulty in detecting the features due to difficulty in distinguishing them, e.g. finding all the faces in a crowd (top right), black and white straight lines on a building looking like text (bottom right).

Informedia: Face and Text Detection

Consider how we might address these problems?

- Look for flesh tones and textures to find faces.
- Look for stone texture as a building rather than text.

Informedia: OCR in Image Text

Informedia: OCR in Image Text

Video OCR from a single keyframe can be difficult, e.g. the fidelity of television images can be quite poor. So it is quite common to misrecognise individual characters.

How to address this problem?

One idea:

- Use multiple keyframes from the same section of video - will have the same letters on the screen.
- Perform Video OCR on each keyframe.
- Then take some vote of the majority for each letter. Hopefully the majority count for each letter will be more reliable than output of a single keyframe.

Informedia: OCR in Image Text

- Check spellings of video OCR with a dictionary, but many caption words are names or jobs titles which may often not be in a dictionary.

  How can we find examples of correct spelling in context of news story?

- One idea:
  - use the “most likely to be correct” character strings as a search query to a text search engine (e.g. online newspaper or newswire service);
  - then check for the presence of the words or similar ones, e.g. in this example we are likely to find stories about “Gerry Adams”, but not “CERRN ADAMS” as detected by the video OCR system in the example.

Video Search

Various forms of query can be used:

- Query by submitting text queries alone and returning back ranked lists of shots.
- Query by submitting text queries and lists of required features:
  - e.g. an outdoor, landscape with text query “aircraft takeoff” - implement as a text search followed by boolean shot filtering according to the specified criteria (e.g. must be outdoor shot).

Video Search

- Query by submitting sample keyframes or drawings of required content:
  - use image retrieval techniques to match queries to the shot keyframes.
- Query for named object, e.g. submit a face and get back shots containing this person.
- Query video specifically e.g. “a green car travelling towards the camera” - needs to include temporal information in the index data.

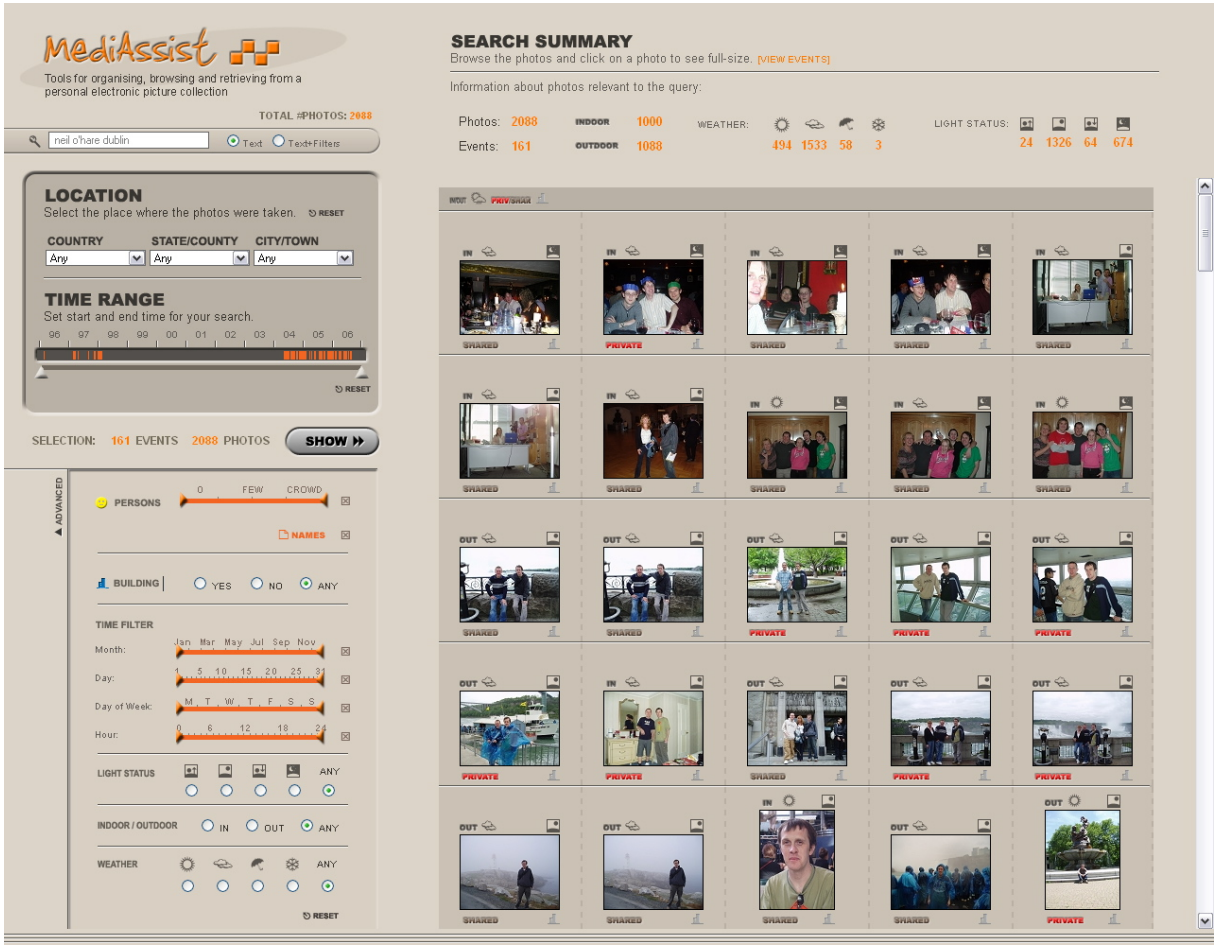
Video Search

- Do people want to browse or search?
  - Browsing is time consuming, but searching may not be accurate enough yet.
  - Best is a combination of searching to narrow the number of videos which need to be browsed to find useful content. Thus video retrieval is typically a two phase process.

  The “human-in-the-loop” - interactively and gradually specifying and refining description of the information need.

Personal Digital Photo Search with *MediAssist*

MediAssist digital photo search with location and time.



Personal Digital Photo Search with *MediAssist*

- Designed for personal photo search.
- Photos uploaded into MediAssist photo search application.
- All images stamped with time and date, plus GPS location.

Personal Digital Photo Search with *MediAssist*

- Photos are grouped into “events” based on time and location.
  - “Event” is photos taken in roughly the same place in fairly quick succession - i.e. it’s a bit approximate, but is quite accurate.
- GPS location is looked up in a gazetteer of place names, e.g. DCU, Glasnevin, Dublin, Ireland.
- Time and date mapped to other values: name of month, day of the week, season (summer, autumn, etc).
- Ambient light condition calculated based on time, date and location - is it dark, light, dusk, etc.
- Prevailing weather conditions looked up online from weather station archive data.

Personal Digital Photo Search with *MediAssist*

Semi-automated annotation of names of people in photos.

- Faces detected in photos.
- Linked together in an event using “body patch matching”, e.g. looks for colours beneath a face - to link the same person between photos.
- Matches faces against models and names in a database.
- User can confirm or correct suggested names for people in photos.
- Corrections are used to update the parameters of the models. The objective being to improve the accuracy of the face identification over time, as more training examples are included.