# Module 11

# VIDEO INDEXING AND RETRIEVAL

# Lesson 38

# Basics of content based image retrieval

## Instructional objectives

At the end of this lesson, the students should be able to:

1. Describe the algorithm for extraction of index keys from JPEG compressed image.
2. Describe the algorithm for computing image similarities based on keys.
3. Extend the concepts of key extraction to video sequences.
4. Outline the methodology for dominant motion estimation for scene structuring.
5. Out line the technique for computing mosaiced images.

## 38.0 Introduction

With the growth of multimedia computing and the spread of the Internet, more and more users can display and manipulate images and would like to use those for an increasing number of applications. A problem is that most image databases are not indexed in useful ways, and many are not indexed at all. It is a formidable task to create an index of images in general and we address only a part of the problem, i.e. the creation of an index that allows retrieval of images similar to a given image presented as a query. We can try to find images that are of the same kind - for example, given an image of a person's face, the aim is to retrieve other facial images from the database.

The concepts of databases management systems have been extended to deal with images. For example the QBIC (Query Based Image Content) system attempts to retrieve image and video using a variety of features, such as colour, texture or shape as keys.

## 38.1 Michael Shneier & Mottalab's approach

This method takes advantage of JPEG coding both to decrease the amount of data that must be processed and to provide the basis for the index keys used for retrieval.

### 38.1.1 **Construction of keys**

The algorithm for selecting random windows is given below:

1. Choose the number of bits, *K* for the index keys. The number of windows used will be 2*K*.

2. Select the window coordinates to tile the image.

3. Determine the window size, as a function of the image size. For compatibility with JPEG 8 x 8 blocks, we clip the window to the smallest multiple of 8 less than this in each dimension.

4. Randomly pair up the window, with the constraint that each window have only one partner.

The algorithm of computing key values for image indexing is given below :

1. Select a set of windows and pair them up.

2. For each pair of windows, allocate a bit in the index key.

3. For each window, compute a vector of features. We take the already computed DCT coefficients in each 8 x 8 block of window as the features and compute the average of each DCT coefficients for all the blocks in the window, giving a vector of 64 feature values.

4. For each feature value and each window pair, compute an index key, giving a vector of index keys, one for each feature, as follows: Compare the value of the first window with that of the second. If the Difference in value is greater than a threshold, assign a 1 to the Corresponding bit; otherwise assign 0.

5. Store the vector of image keys as the index to the image.

**38.1.2 Image based retrieval**

The algorithm for computing image similarity is as follows:

1. Construct a vector of keys for the index image using the same arrangement of windows and the same features that were used to construct the indices for the images in the database.

2. Compare the key with the keys of all the images in the database, on a bit by bit basis.

3. Compute the degree of match as the sum of all bit positions that are different.

4. Repeat the match computation for each of the features in the key vector, and sum the results for all the features

5. The total number of differences is the measure of similarity.

## 38.2 Video retrieval

The simplest way to represent video for browsing and querying is through key in a short, typically the first, middle, last frame or a combination of these. Thereafter, QBIC like queries can be performed.

Sawhney and Ayer presented techniques for automatic decomposition of a video sequence into multiple motion models and their layers of supports, which together constitute a compact description of significant scene structure.

This includes:

a) Separation of the dominant background scene from moving objects.

b) Representation of the scene and moving objects into multiple layers of motion and spatial support. Furthermore, the motion based decomposition of videos can be used to create compact views of numerous frames in a shot by video mosaicing.

There are two major approaches to the problem of separating image sequences into multiple scene structures and objects based on motion.

One set solves the problem by letting multiple models simultaneously compete for the description of the individual motion measurements, and in the second set, multiple models are fleshed out sequentially by solving for a dominant model at each stage.

## 38.3 Simultaneous multiple motion estimation

Essential idea : Iteratively clustering motion models computed using pre-computed dense optical flow. It is main draw backs are:

a) In computing optical flow, we make soothes assumptions that can distort the structure of image motion.

b) Clustering in the parameter space is generally sensitive to the number of clusters specified.

### 38.3.1 Dominant Motion Estimation:

Sequential application of dominant motion estimation methods have been proposed for extracting multiple motions and layers.

Ayer *et al* combined intensity-based segmentation with the motion information. However, they noticed that even with the use of robust estimations, the

sequential dominant motion approach may be confronted with the absence of dominant motion.

Robust estimation of a motion model using direct methods :

Given two images, their motion transformation is modeled as

$$I(\mathbf{p},t) = I(\mathbf{p} - \mathbf{u}(\mathbf{p};\theta), t-1)$$

where **p** is the 2-D vector of image coordinates and **u**(**p**;θ) is the displacement vector at **p** described using a parameter vector θ .

- 2- D global parametric model: a low dimensional θ describes the motion.
- 3-D global parametric model: a low dimensional global parameter + projective depth part.

In order to compute motion of varying magnitudes, the images are represented at multiple scales using Gaussian or Laplacian pyramids.

In the *M*-estimation formulation, the unknown parameters are estimated by minimizing an objective function of the residual error. In particular, the following minimization problem is solved:

$$\min_{\theta} \sum_{i} \rho(r_i;\sigma), \quad r_i = I(\mathbf{p}_i,t) - I(\mathbf{p}_i - \mathbf{u}(\mathbf{p}_i;\theta), t-1)$$

where, $\rho(r;\sigma)$ is the objective function defined over the residuals r, with a given scale factor σ and where *i* is the index of *i* th pixel.

$$\rho_{ss}(r;\sigma) = \frac{1}{2}\frac{r^2}{\sigma^2}, \qquad \rho_{GM}(r;\sigma) = \frac{\frac{r^2}{\sigma^2}}{1+\frac{r^2}{\sigma^2}}$$

## 38.4 Video Mosaicing with dominant 2 D Motion Estimation

In many real video sequences where the camera is panning and tracking an object, a panoramic view of the background can be created by mosaicing together numerous frames with warping transorms that are the result of automatic dominant motion computation. The 2-D motion estimation algorithm is applied between consecutive pairs of frames. Then a references frame is chosen and all the frames are warped into the coordinate system of the reference frame. This process creates a mosaiced frame whose size, in general, is bigger than the

original images; parts of the scene not seen in the reference view occupy the extra space. Temporal filtering of various warped images in the mosaic frame's coordinate system creates the mosaiced image.