Category: Multimedia Technology

Recent Progress in Image and Video Segmentation for CBVIR

Yu-Jin Zhang
Tsinghua University, Beijing, China

INTRODUCTION

A simple search from EI Compendex by using the term "image segmentation" only in title field could produce around 5000 records (Zhang, 2006). However, as no general theory for image segmentation for different application domains, particular algorithms have been developed. The domain of Content-Based Image Retrieval (CBIR) is such a typical example, where many specific techniques have been proposed. An introduction focused on research works before 2004 can be found in Zhang (2005). This paper is an up-to-date and extended version from CBIR to CBVIR (Content-Based Visual Information Retrieval) by including CBVR (Content-Based Video Retrieval), which focused on the progress in last 3 years, and especially on video segmentation.

BACKGROUND

A Formal Definition of Image Segmentation

Image segmentation is the first step and also one of the most critical tasks of image analysis. It is often described as the process that subdivides an image into its constituent parts and extracts those parts of interest (objects).

A formal definition of image segmentation, supposing the whole image is represented by $f(x, y)$, and $f_i(x, y)$, $i = 1, 2, \ldots, n$ are disjoint non-empty regions of $f(x, y)$, consists of the following conditions (Fu, 1981):

1. $\bigcup_{i=1}^{n} f_i(x, y) = f(x, y)$;
2. For all i and j, $i \neq j$, there exits $f_i(x, y) \cap f_j(x, y) = \varnothing$;
3. For $i = 1, 2, \ldots, n$, it must have $P[f_i(x, y)] = TRUE$;
4. For all $i \neq j$, there exits $P[f_i(x, y) \cup f_j(x, y)] = FALSE$; where $P[f_i(x, y)]$ is a uniformity predicate for all elements in $f_i(x, y)$ and $\varnothing$ represents an empty set. Considering the real situation in practice, the following condition can be added:
5. For all $i = 1, 2, \ldots, n$, $f_i(x, y)$ is a connected component.

In the above conditions, condition (1) points out that the summation of segmented regions could include all pixels in an image; condition (2) points out that different segmented regions could not overlap each other; condition (3) points out that the pixels in the same segmented regions should have some similar properties; condition (4) points out that the pixel belonging to different segmented regions should have some different properties; and, finally, condition (5) points out that the pixels in the same segmented region are connected.

Definition Extension to Video Segmentation

If a 2-D still gray level image is represented by $f(x, y)$, then its extension to 3-D moving images or sequences of images (video) can be represented by $f(x, y) \Rightarrow f(x, y, t)$. In video domain, two kinds of segmentation can be distinguished: *spatial segmentation* and *temporal segmentation*. In *spatial segmentation*, each frame of $f(x, y, t)$ can be denoted as $f_t(x, y)$, which is a 2-D still image and the above formal definition for image segmentation can still be used.

The *temporal segmentation* of video can be defined as follows. Given a video sequence $f(x, y, t) = \{f_1(x, y), f_2(x, y), \ldots, f_i(x, y), \ldots, f_n(x, y)\}$, the k-th partition of $f(x, y, t)$ can be denoted as $g_k(x, y) = \{f_i(x, y), f_{i+1}(x, y), \ldots, f_{i+ik-1}(x, y)\}$, where i_k is the number of frames in the k-th partition, and $\sum_{k=1}^{m} i_k = n$. The formal definition of temporal video segmentation consists of the following conditions:

1. $\bigcup_{k=1}^{m} g_k(x, y) = g(x, y)$;
2. For all k and l, $k \neq l$, there exits $g_k(x, y) \cap g_l(x, y) = \varnothing$;
3. For $k = 1, 2, \ldots, m$, it must have $P[g_k(x, y)] = TRUE$;
4. For all $k \neq l$, there exits $P[g_k(x, y) \cup g_l(x, y)] = FALSE$.

Compared with the definition of image segmentation, the corresponding condition (5) has already been included in the definition of $g_k(x, y)$. In other words, the frames in the same shot are connected in time.

MAIN THRUST

In recent years, many researches on image and video segmentation for CBVIR are carried out. Two of them are described with some detail in the following, some others are just briefly indicated.

Color Image Segmentation in Feature and Image Spaces

In CBVIR, color information plays an important role. Early retrieval algorithms for CBVIR are often based on the color information of image or object. Many current retrieval algorithms are still using color information to derive semantic description. Therefore, efficient color segmentation techniques are critical for CBVIR.

One color segmentation technique based on *watershed* and feature space analysis is described below. This technique made the combination of *watershed* transform and feature space analysis.

In most cases, the *watershed* algorithm is applied on image domain (usually on the edge image). It focuses on local color feature instead of global color distribution. In edge image, the local minima exist in the interior of objects and high altitude appears on the boundary of objects. After a flooding process, dams (*watershed* lines) will be constructed on object boundary and different objects are separated. In this way, it captures only information of local color feature instead of global color distribution.

In feature space, one obstacle of segmentation is the difficulty relies on color clustering. Researchers have noticed (Park, 1998; Pauwels, 1999) that color distribution in 3-D color space cannot be well approximated by the traditional parameter based clustering algorithm (such as *K*-mean model or Gaussian mixture model). For example, *K*-mean is unable to handle unbalanced or elongated clusters. Gaussian mixture model is not appropriate for cluster with irregular shape. One example is shown in Fig. 1. The original image PEPPERS is in Fig. 1(a), its pixel distribution in color RGB space is projected onto 2-D plane as in Fig. 1(b) and its pixel distribution in color $L^*a^*b^*$ space is projected onto 2-D plane as in Fig.1(c). In Fig. 1(b), the distribution has irregular shapes, some clusters are sharp and compact, and some are flat. In Fig. 1(c), the clusters seem more salient, but it is still hard to get the boundary between clusters with parametric models.

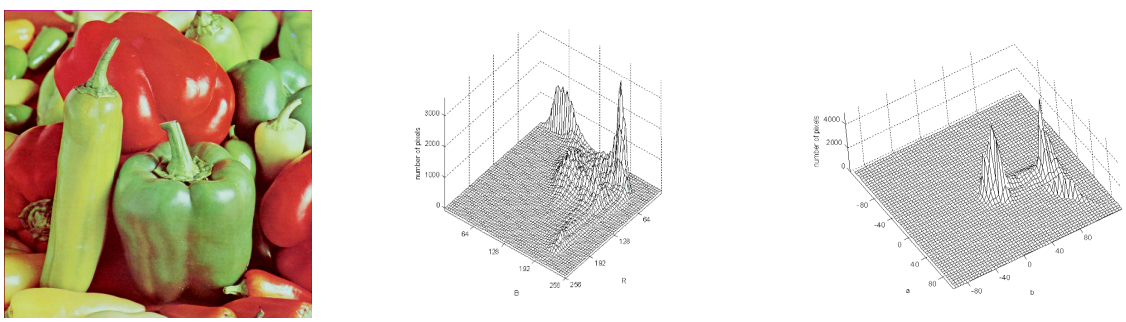
To solve the problem, the *watershed* algorithm is applied on a 3-D $L^*a^*b^*$ histogram $H(x, y, z)$ to capture the feature space information. A labeling process with the following steps is used to cluster the color histogram (Dai, 2006):

1. Get the reverse histogram $H'(x, y, z) = -H(x, y, z)$ $(0 \le x < u, 0 \le y < v, 0 \le z < w)$.
2. Get all local minimum of the reverse histogram H', label them as 1, 2, 3, …, m.
3. Find the unlabeled bin in H' with minimum value and label it according to its neighbors:
   i. If more than one label appears in its neighborhood, it is a "dam" bin, and it will be labeled as 0.
   ii. If else, label it the same as its labeled neighbor.
4. Go to step (3) until all non-zero bins are labeled.

After obtaining the *watershed* in the color histogram, the results can be brought back to the image space. The following post-process steps are used to get continuous homogeneous regions with meaningful size.

1. Get all pixels with corresponding color bins labeled

Figure 1. Pixel distribution of image PEPPERS in color space



(a) *(b)* *(c)*