Contents

1 Definition

Object Recognition is given an image and knowledge of models, classifying the objects as one of models of objects and figuring out the orientation of them in the image.

1. Which objects are there in the image? (classification)
2. What are the poses of the objects in the image? (Rotation and Translation w.r.t. camera)

2 What makes Object Recognition possible?

Each object has characteristic such as shape, color, texture and motion. These characteristics will not be precisely identical, yet it will give us important cues so as to enable us human to recognize objects. Think of movement of human mouth when one speaks. This gives an useful cue to recognize a mouth.

3 What makes it hard?

Object recognition is still a largely unsolved problem. Why?

1. Intrinsic variability in each class
   Not every zebra is the same, yet we want to recognize them as members of the same class. Very little has been done to account for this. Some sort of learning and statistical approach is needed.

2. Pose variability
   We can deal with this most readily using geometry.

3. Illumintation variability
   Blindly matching from a set of templates will not work because two different scenes will likely to have different illumination.

4 Possible Approaches

There are three possible approaches. Let's take a look at them one at a time.

1. Segment first, recognize later

2. Recognize first, segmentation is immediate

3. Recognize and segment simultaneously

4.1 Recognize first, segmentation is immediate

Alignment method [Hutlenlocher, Ullman]: using Shape alone, no color and texture
Basic idea:

- Extract 'corner' type features in projection

- From small set of matching corners, and matching object calculate the pose and predict the remaining corners

- Compute score for every objects

- Pick highest scored object to be most likely matching

How can you first know which set of points corresponds to which set of corners in the object? We will use **Hypothesize and Test** approach. We will first hypothesize the correspondence between a set of points and set of corners in the object, and verify if the hypothesis was good. This would take $O(m^3 n^3)$ to choose the right combination of points and corners because there are roughly $O(m^3)$ ways to choose 3 points from m points, and $O(n^3)$ ways to choose 3 corners from n corners. Verification takes $O(m \log n)$ to check whether given hypothesis was good. Thus total running time of this approach is $O(m^4 n^3 \log n)$. Pose estimation can be done as well after corresponding points are known.

- Notice this algorithm will first look for a chair, then for tiger, then for pencil, etc. in the given image sequentially. This can be made efficient with proper data structures such as indexing scheme.

- The algorithm fails when there is no explicit corners in the 3D object. Thus, cylinder or sphere will not be recognized properly. Also richly textured objects such as tiger will have unnecessarily large n.

- One advantage of this approach is that there is no low level segmentation and grouping involved. It can just pull of the objects from the image without low level operations.

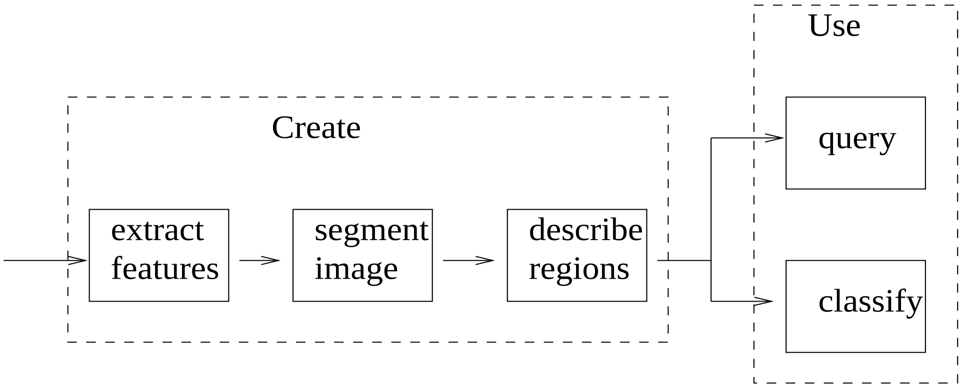

Figure 1: Blob world diagram

4.2 Segment first, recognize later

Let's look at approaches using color and texture alone, and no color. Why? Because shape is hard to use in general setting because of intrinsic and pose variability. **Content based querying** is querying the picture for a specific content, for example looking for a picture with a zebra. In this scenario, classical approaches such as alignment do not work.

QBIC (Query by Image Content) by IBM

QBIC uses color histograms to represent an image's content. Color histogram is a frequency plot of number of pixels appearing in the image with each quantized color value. Then the histograms are compared by using cityblock distance, i.e. $\Sigma_n |H_1(n) - H_2(n)|$. The reasoning behind QBIC is that similar kind of scenes will have similar color distribution, thus similar color histogram. But this doesn't describe the individual objects at all.

Blob World by U.C. Berkeley

Blob is an arbitrary region in an image. What Blob World approach does is to pre-process the images so as to extract features and segment the images. Each blob, a coherent region of image is then described in terms of color and texture. When a user comes along and inputs a query, the query is processed dynamically to find the pictures with best matching blobs and returns them (refer to the above diagram).

The system has been implemented and have shown about 50rate. This performance is reasonable for naive users who wants to find pretty pictures of animal. More detail can be found at the following URL: http://elib.cs.berkeley.edu/photos/blobworld

4.3 Recognize and segment simultaneously

Very little has been done to explore this method in vision context. In speech recognition, HMM (hidden Markov model) is applied very successfully. But it's not clear how this can be extended to computer vision.