CS 585 Image and Video Computing

Project Proposals

Don Johnson

Email: [donj@bu.edu](mailto:donj@bu.edu)

**I. Goal:** Classify and count items in stainless steel flatware set.

I propose to create a system that can count stainless steel flatware such as forks, knives and spoons in an image. The system will be trained to recognize the items in training data, and the object descriptors will be persistent so that when presented with an image containing various numbers of forks, knives and spoons of the same kind present in the training data, it can count them accurately.

**Primary problems:**

a. The objects are texture-less. I want to use cheaper, pattern-free silverware because I want to focus on morphology as opposed to texture and pattern.

b. Any keypoint/image patch approach will suffer from the aperture problem

**Additional problems**:

a. Occlusion is the most difficult problem. Objects underneath other objects will be difficult to count. The may be more than even two layers of objects. Each count for a class of object (e.g. forks) must have an associated probability and confidence interval as opposed to an absolute count.

b. I will include some calibration scale on the surface holding the objects so size and number of pixels can be correlated. However, the objects will have to be registered because then can be rotated and translated in any way. When we start piling them up, they can also present an oblique view to the camera if they are not laying flat, but propped up at an angle to the image plane by other objects underneath.

c. Change the background to a reflective surface to see if the reflections can be filtered out of the image.

d. Add forks with flat tines and see if they can be distinguished from forks with curved tines. This may necessitate using stereoscopic images.

**Source of data**: Images taken myself with a Sony SLR.

**Algorithms**

Assume every operation performed on the image data has already been done to the training data, labeled and stored in a database in some type of descriptor

**Machine Learning**

Might want to create n-dimensional descriptor based on output of Stages 1, 2 results and use k-means clustering to classify objects.

**Stage 1 Classification base on blob characteristics**

Distinguishing objects: use MISR to detect blobs. Calculate area, perimeter, elongation, and compactness and circularity. Normalized pixel count via calibration scale before doing calculation. Create a correlation matrix between training set values and detected values for blob. Calculation probability that each blob is each respective object in training set

Anything not having close correlation to some set of blob characteristics is probably a “combined” object because of occlusion. Set the object(s) aside so the parts can be separated and re-input in the analysis process.

**Stage 2**

Detect edges using Canny.

Do a closing operation and/or Hough transform

Do contour matching.

Using Szeliski Ex 4.9

Walk contour and create list of (xi, yi, si) triplets using arc-length formula for si.

Resample list onto a regular set of (xj, yj, sj) using linear interpolation

Compute x, y averages and subtract from all triplets.

Resample piecewise-linear function onto length-independent set of samples

Compute Fourier transform of curve treating each (x,y) pair as complex. This decomposes each curve into a set of coefficients representation each curve as a linear composition of sinusoids (pretty cool!). The descriptor stored in the database will be the number (order) and magnitude of the coefficients. Have to come up with some “fuzzy” way of correlating the descriptors. The coefficients will be rotation, translation and scale (?) invariant. They will not be invariant for items that are not orthogonal to the camera optical axis. Don’t know what to do about that.

Also, will classify based at sharp and rounded corners: top of tines on a fork, tip of a knife and the bottom and top of spoon.

**Stage 3**

Work on occluded composite objects remaining from Stage 1 analysis.

Iterative process

Use spline and/or snake to fit curves to lines in object. Look for closed blobs in object and remove one.

Feed blob back into Stage 1; check for match. Next, feed it to Stage 2 and look for partial match base on curves.

Repeat by removing next blob in object.

For example, say a fork is lying on top of a spoon. It will be extracted as a blob, fed back into Stage 1 and 2. Stages 1 and 2 will be identifying it as a fork. Now, you will be left with two pieces of a single spoon with a section missing where the fork was. Individually, the spoon pieces will fail to match anything in Stage 1, but both pieces will have some curve matching in Stage 2. You might also be able to do some region merging and come up with a blob that will be matched by feeding it back into Stages 1 and 2.

**Stage 4**

Create confusion matrix from having human do actual count and analyze results. Refine algorithms and/or classification scheme.

**II Goal**: Tracking and classifying tropical fish

I think that detecting motion, analyzing color and texture would be very important in tracking and classify tropical fish.

Even when the fish are sleeping, the fins flap to maintain equilibrium. I would analyze the frequency and even look at the probability distribution function with respect to time of the fin motion. This distribution would be different that the rhythmic waving on plants in water current.

Detecting blobs in all three color spaces and looking at color composition and ratio could help classify the type of fish.

SIFT would also be useful as long as you could extract “good” images in a video sequence where the fish was swimming perpendicular to the camera axis. This would generally be when the blob area was largest in the sequence. Optical flow techniques could be used because swimming patterns might be distinctive for different type of fish.

**III Goal**: Detect people who are not wearing their seat belts and use the results of someone else’s license plate detection process to send the scofflaws a very expensive ticket. This is Massachusetts after all.

Camera position would be very import because the best perspective would be straight on through a windshield.

First, I would use Viola-Jones, Haar filter to detect the face. I would use the face, and bridge of the nose to come up with an orientation line and centroid.

Next, I would look under the face for contour of the shoulder. Finally, relative to face and shoulder, face centroid and orientation line, I would look for angled parallel lines representing the presence of a safety belt.