CS365: ARTIFICIAL INTELLIGENCE

February 20

2012

Kanish Manuja
Vempati Anurag Sai

Project Proposal

Learning Human Actions

Introduction: We intend to design an intelligent agent that can learn realistic actions performed by humans. For this we use the dataset of various movie clips that has been used in [1]. Our project follows closely along the work in [1].

Motivation

Our Project gets its motivation from successful object detection or scene categorization from unlabeled static images. For a detailed review of 2D descriptors see [4]. But, it is a challenging task for computers to classify different actions in video sequences because of the photometric, geometric variance and background cluttering. Various techniques to overcome these problems are discussed in [3]. It is very useful for a variety of tasks, such as video surveillance, object-level video summarization, video indexing, digital library organization, etc.

HISTORY

A substantial amount of work is being done on the problem of learning human actions using various methodologies in the recent times.

Bobick and Davis (2001) used motion history images that capture motion and shape to represent actions. They have introduced the global descriptors *motion energy image* and *motion history image*, which are used as templates that could be matched to stored models of known actions. Their method depends on background subtraction and thus cannot tolerate moving cameras and dynamic backgrounds. Efros et al. (2003) perform action recognition by correlating optical flow measurements from low resolution videos. Their method requires first segmenting and stabilizing each human figure in the sequence, as well as further human intervention to annotate the actions in each resulting spatial-temporal volume. In the work by Song et al. (2003) and Fanti et al. (2005), feature points are first detected and tracked in a frame-by-frame manner. Multiple cues such as position, velocities and appearance are obtained from these tracks. Again dynamic backgrounds could not be handled efficiently.

In spite of the existence of a fairly large variety of methods to extract interest points from static images (Schmid et al. 2000), less work has been done on space-time interest point detection in videos. Laptev (2005) presents a space-time interest point detector based on the idea of the Harris and Förstner interest point operators (Harris and Stephens 1988). They detect local structures in space-time where the image values have significant local variations in both dimension.

Procedure:

Hereafter, we will be considering a video as a set of gray scale images in 3rd dimension (time).

The project involves the following main stages:

1. Identifying Space-Time interest points (STIP) in a video and collecting cuboid prototypes around them.
2. Transformations on the cuboids and building feature descriptors.
3. Using K-means clustering algorithm to find clusters of features found in videos sampled from the dataset.
4. Building Spatio-temporal Bag of features (BoF).
5. Using State Vector Machine (SVM) to train the agent to classify various videos.

These stages can be summarized as follows:

1. **STIP:** The aim is to obtain robust, stable and well-defined image features for object tracking and recognition. Corner detection was being used for many years in the past as interest points. We will be using Harris affine detector for our purpose. It relies heavily on both the Harris measure and a Gaussian scale-space representation.

   The basic idea is as follows (assuming a 2-D image for simplicity):
   The Harris corner detector algorithm relies on a central principle that, at a corner, the image intensity will change largely in multiple directions.

   Let this image be given by *I*. Consider taking an image patch over the area (*u*,*v*) and shifting it by(*x*,*y*). The weighted *sum of squared differences* (SSD) between these two patches, denoted *S*, is given by:

$$S(x,y) = \sum_u \sum_v w(u,v)\ \left(I(u+x, v+y) - I(u,v)\right)^2$$

   By using Taylor expansion for ***I* (*u* + *x*, *v* + *y*)** and after some algebraic manipulations, S(x, y) can be written in a matrix form.

$$S(x,y) \approx \begin{pmatrix} x & y \end{pmatrix} A \begin{pmatrix} x \\ y \end{pmatrix},$$

   Where A is given by,

$$A = \sum_u \sum_v w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix},$$

   I_x and I_y be the partial derivatives of *I*

   If A has 2 large eigenvalues, it is a corner point. But since evaluating eigenvalues takes a lot of computations, an alternative was proposed. A corner point is a point where the function 'Mc' is locally maximum and above some threshold.

$$M_c = \lambda_1 \lambda_2 - \kappa \left(\lambda_1 + \lambda_2\right)^2 = \det(A) - \kappa\ \mathrm{trace}^2(A)$$

So, it's enough to compute determinant and trace of 'A' to find corners. A Gaussian scale-space representation of the image is used for handling image structures at different scales. On similar lines, this can be extended to 3-dimensional case. Points either corresponding to corners in a frame or sudden changes in temporal dimension are detected by this method. The detection of the points also depends on the scales used for the gaussian. The theory and the considerations that arise while dealing with temporal dimension have been discussed in [2].

2. **Feature Descriptors:** The spatial-temporal interest points have important features around them that help us in differentiating between videos of different classes. So, we build cubiod prototypes around the STIPs to collect those features. Several transformations are performed on these cuboids to make them invariant to difference in lighting, appearance, motion etc. yet retaining their discriminative power. These might include normalization of pixel values, brightness gradient, windowed optical flow etc. Once the transformations have been performed, a feature vector is generated by stretching out the pixel values of the cubiod into a vector (sensitive to perturbations) or histogramming the values (loss of positional information). We will also use Isomap to reduce the dimensionality of the feature vector. Refer [3] for additional information. We would be using HoG (Gradient transformation + Histogramming) and HoF (Optical flow transformation + Histogramming).

3. **Clustering:** Although two instances of the same behaviour may vary significantly in terms of their overall appearance and motion, many of the interest points they give rise to are similar. So, though the number of possible cuboids is unlimited, the number of different "types" of cuboids is limited. A huge number of samples are obtained from the dataset and clustered using K-means clustering algorithm to find such "different" features in human behaviour. Each behaviour might be having many such features and each feature might be present in many behaviours. But, the "*distribution*" of these features varies significantly for behaviours belonging to different classes.

4. **Spatio-temporal Bag of features:** BoF assigns each feature to nearest vocabulary word. Then, we compute a histogram (similar to "distribution function") of visual word occurrences in the space-time for each video. This histogram can be used to distinguish a particular behaviour from others. Behaviours belonging to same classes have close resemblance in their histograms.

5. **State Vector Machine:** For the classification problem, we will train a non-linear SVM with multi-channel kernel on large data.

References:

[1] Ivan Laptev, Marcin Marszałek, Cordelia Schmid, Benjamin Rozenfeld. Learning realistic human actions from movies. IEEE Conference on Computer Vision & Pattern Recognition, 2008.
[2] I. Laptev. On space-time interest points. *IJCV*, 64(2/3):107–123, 2005.
[3] P. Doll´ar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005.
[4] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In CVPR, pages II: 257-263, 2003.