KSU Willie in Semantic Vision Challenge

Dr David Gustafson, Aaron Chavez, Michael Marlen, Andrew King, Alejandro Alliana, and Ondrej Linda

Computing and Information Sciences
Kansas State University
Manhattan, KS 66506

Abstract

The KSU Willie entry in the Semantic Vision Challenge will use a variety of classifiers, some standard classifiers and some newly developed classifiers, to learn the classification of images downloaded from the web. KSU Willie will use those classifiers to identify objects in the environment.

Introduction

The Kansas State University entry into the AAAI 2007 Semantic Vision Challenge consists of a Pioneer P3AT robot (see figure 1) running Windows 2000, scalable client/server software architecture, a path-planning package, and a set of image classifiers that will be trained on images downloaded from the internet.

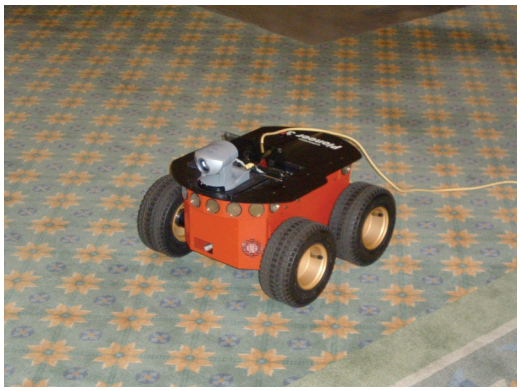

Figure 1: KSU's P3AT robot

Classifiers

Our approach to the semantic vision challenge is to develop a set of classifiers based on different characteristics of the objects. These include a number of standard classifiers. Additionally, we have developed some unique classifiers. All of these classifiers will be trained autonomously on images downloaded from the internet. The resulting trained classifiers will be used to identify the objects in the environment. The effectiveness of the classifiers on the training set will be used to determine which classifier or set of classifiers will be used for each object in the environment.

The Marlen Perceptual Model

The Marlen Perceptual Model provides a unique way of looking at objects. MPM is based on pairs of colors. The restriction on the pair is that the first color must be adjacent to the second color and that the colors are sorted from least to greatest. Learning is performed based upon the colors seen in the training set of images. MPM works well on objects from never- before-trained-on-perspectives if the object has a similar color theme across the object.

The MPM, given a strong set of training images, works better than approaches that are based on shape or explicit information such as from SIFT. To be a strong training set of images, there are two criteria. First, the training images must have a similar color theme (high correlation between color patterns) for the positive training set and, second, the negative training set examples must contain background information contained within the positive training set.

Probabilistic Feature Bagging

Probabilistic feature bagging uses salient feature detection and extraction, features as codewords, and two stages of probabilistic analysis (one for classification and one for image localization) of features. The first stage of analysis extracts SIFT and SURF feature vectors from training images (both positive and negative examples).

This list of features is used to generate a codebook of codewords via k-means clustering. We use a codebook

with 1000 codewords. The features for each training image is resolved to a codeword, then a histogram for the image that represents of the frequency of codewords is produced. A Bayesian learning system learns what codeword frequencies are representative of the training images. Also, the raw features themselves are annotated with the class of the image they were extracted from and are fed to their own Bayesian learning system.

Image classification takes place in two stages. In the first stage, features are extracted from the image and used in conjunction with the codebook to generate a histogram of the image. A Bayesian classifier determines what the most likely class of the image is. If the image gets classified as a positive example, each of its raw features are passed into the secondary Bayesian system. If the feature is off the class of the image, then the region of that feature is recorded. The positive regions are merged in order to produce bounding boxes around the objects in the image.

Geometric Feature Descriptor

The geometric descriptor attempts to recognize objects using basic shapes/outlines. Initially, an image segmentation algorithm divides the picture into regions. We must then find outlines of potential objects by finding the edges between "foreground" and background". If a region is sufficiently large, it is labeled as background. Then, we need only locate the regions directly adjacent to the background.

Once the regions of interest are located, we must classify them. We first find the prominent lines in the image with a Hough transform. Then, we convert them to line segments. Next we group each pair of lines and convert them into a descriptor.

So, each descriptor describes one pair of line segments. The parameters for the descriptor are ratio, orientation, angle, and distance. The ratio describes the length of the longer segment, relative to the length of the shorter. The orientation describes the angle of the longer segment relative to the image itself. The angle describes the angle between the two segments. The distance describes an approximation of the distance between the two segments.

The training algorithm learns which approximate values to expect for ratio, orientation, angle, and distance. In the live phase, it attempts to find matches in the test images.