
Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02), Vol. 5
Lipo Wang, Jagath C. Rajapakse, Kunihiko Fukushima, Soo-Young Lee, and Xin Yao (Editors)

A RECOGNITION SYSTEM THAT USES SACCADES TO DETECT CARS FROM REAL-TIME VIDEO STREAMS

*Predrag Neskovic, Leon N Cooper**

Brown University
Physics Department and
Institute for Brain and Neural Systems
Providence, RI 02912, USA

David Schuster †

Yale University
Physics Department
New Haven, CT 06520, USA

ABSTRACT

In this work we present a system for detection of objects from video streams based on properties of human vision such as saccadic eye movements and selective attention. An object, in this application a car, is represented as a collection of features (horizontal and vertical edges) arranged at specific spatial locations with respect to the position of the fixation point. During the recognition process, the system efficiently searches the space of possible segmentations by investigating the local regions of the image in a way similar to human eye movements. In contrast to motion-based models for vehicle detection [3, 4], our approach does not rely on motion information, and the system can detect both still and moving cars in real-time.

1. INTRODUCTION

Identification of vehicles from video streams is a challenging problem that incorporates several important aspects of vision including: translation and scale invariant recognition, robustness to noise and occlusions and ability to cope with significant variations in lighting conditions. In addition, the requirement that the system work in real-time often precludes the use of more sophisticated but computationally involved techniques.

The problem of vehicle identification from video streams has been widely addressed in computer vision literature [3, 6, 5, 4]. Very often, an underlying assumption is that the vehicles are moving and motion information is used to segment the image into moving regions and a static background. Based on its overall size and shape, a region can then sometimes be recognized as a vehicle even without a detailed description. Furthermore, motion information can reduce the computational complexity since only the regions that contain motion have to be analyzed. However, in many situations, motion information is not available or is insufficient, and other ways of dealing with computational complexity and segmentation problems have to be used. Biologically inspired vision systems may provide one such solution.

Due to the structure of the eyes, the human visual system does not process the whole visual input with the same resolution. The region of the scene that is perceived with the highest quality is the one that projects to the fovea, an area of the retina corresponding to only about the central 2 degrees of the viewed scene. The regions that are further away from the fixation point are perceived with progressively lower resolutions. The visual system overcomes this limitation by making rapid eye movements, called *saccades*. Human recognition is therefore an active process of probing and analyzing different locations of the scene at different times and integrating information from different regions.

Biologically-based recognition systems have been proposed for various applications such as face recognition, handwriting recognition and vehicle detection [2]. An approach to object recognition that is based on human

* Supported in part by the Army Research Office.
† The author performed the work while at Brown University.

saccadic behavior is proposed in [5]. While this model does capture properties of saccadic behavior, it represents an object as a fixed sequence of fixations.

In this paper we propose a new model for object recognition that employs properties of human vision such as selective attention and saccadic eye movements. This work is an extension of our previous work [8, 7] that was applied to segmentation and recognition of one-dimensional objects, handwritten words. In this work we show that our model can be successfully applied to recognition of two-dimensional objects, such as cars. In contrast to motion-based models for vehicle detection [3, 4], our approach does not rely on motion information, and the system can detect both still and moving cars in real-time.

2. OBJECT REPRESENTATION

Inspired by the properties of human vision, such as saccadic eye movements and foveal vision, we model an object as a collection of features of specific classes arranged at specific spatial locations with respect to the fixation point. For example, if an object is a word, then the features can be the letters, and if an object is a car the features can be the edges of different orientations and sizes.

Most features, within objects of the same class, vary in shape and size. A striking example is the variation of shapes and widths of letters that represent the same dictionary word. Even within the class of a rigid object, such as a car, the variations can likewise be considerable. As a consequence of the uncertainty associated with the size of each feature, the region of space containing the feature that is close to the fixation point is much smaller than the region of space containing the feature that is further away from the fixation point. This uncertainty of feature locations will in turn determine the strength of the interaction among the features (nearby features will have a larger influence on the feature located on the fixation point than the features that are further away) and will determine the sizes and distribution of the receptive fields of the units of the network.

Interaction among the features - the role of context. Due to the local information contained in a given region (a section of the pattern), its interpretation is inherently very ambiguous. However, including the information from the neighboring regions often removes that ambiguity.

Let us denote the probability of finding the feature f_i within the region R_i as $p(f_i \in R_i) = d_i$ and the probability of detecting the feature f_j within the region R_j as $p(f_j \in R_j) = d_j$. In the rest of the paper, we will always reserve the subscript i to denote the region centered at the fixation point - the central region. Furthermore, let us denote the sizes of the regions R_i and R_j as $S(R_i)$ and $S(R_j)$ respectively. If the average size of the feature that corresponds to the region R_j is $S(f_j)$, then the number of possible locations for the feature f_j within the region R_j is proportional to $S(R_j)/S(f_j)$ and the amount of the overlap among feature locations. Assuming that all locations are equally likely, the probability of finding the feature f_j at any of those locations is $const * S(f_j)/S(R_j)$. Since all the measurements are done with respect to the location of the central feature, this is also the probablity of finding the feature f_j anywhere within the region R_j given the location of the central feature, $p(f^j \in R_j | f^i \in R_i) = p(j|i) = const * S(f_j)/S(R_j)$. The probability of detecting the feature f_i in the region R_i (without any context) is d_i. The term $p(i|j)d_j$ represents the context (for the i^{th} feature) provided by the j^{th} feature. The probability that the feature f_i, represents part of some object is then given as $d_i p(i|j) d_j$. Similarly, the central feature provides context for the feature f_j, and the probability that the feature f_j represents part of the object is given by the same expression, since $p(j) = p(i)$, and therefore $p(i|j) = p(j|i)$.

3. THE ARCHITECTURE OF THE NETWORK

In this section we will describe the architecture of the network that represents one object (in our case a car). The first layer of the network consists of feature detectors whose receptive fields completely cover the input image. Their output, the probability that the feature to which they are selective is within their field, is supplied to the next layer of units called *simple units*. A simple unit is selective to only one feature and is invariant

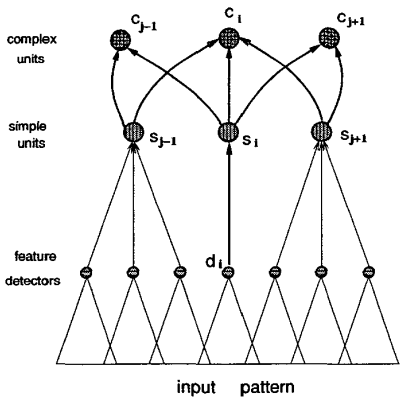

Figure 1: The network architecture.

to the location of the feature within its receptive field. The simple units are divided into groups, each group representing an object from one point of view. Since we assume that the fixation point has to be on one of the features that means that there are as many groups of simple units as there are features. Let us now consider one group of the simple units. It consists of the central unit, the one that is located above the fixation point and the surrounding units. The size of the receptive field of the central unit is the smallest compared to other simple units, and the sizes of the receptive fields of the simple units increase with their distance from the central unit. The sizes of the receptive fields of the simple units are designed is such a way as to accommodate the uncertainties associated with locations of the features with respect to the fixation point.

The output of a simple unit is given as

$$s_k = max_{\vec{r} \in R_k}(d^k(\vec{r})), \tag{1}$$

where $\vec{r}$ is the location of the feature detector that is selective to the k^{th} feature and R_k is the receptive field of the k^{th} simple unit. Therefore, a simple unit outputs the probability that a feature that it is selective to is somewhere within its receptive field. The next layer of the units, called *complex units*, incorporates contextual information. The complex unit that receives input from the central simple unit outputs the probability that the region R_i (or the feature it contains) now represents part of the object

$$c_i = d_i \frac{1}{N-1} \sum_{j=1, j \neq i}^{N} p(i|j) s_j, \tag{2}$$

where N represents the number of features in the object. This means that the detection of the central feature is now viewed within the *context* of all the other features of the object. Similarly, the j^{th} complex unit that receives input from the j^{th} simple unit views the j^{th} feature within the context of the central feature

$$c_j = d_i p(j|i) d_j. \tag{3}$$

According to our model, each local region can represent an object with different confidence. The probability that the collection of all the regions that contain object features represents the object from the point of view of the i^{th} feature is captured by the *object unit*

$$o_i(object|fixation\ point\ i) = \frac{1}{N} \sum_{k=1}^{N} c_k, \tag{4}$$

where the index k goes through all the complex units, the central (i) and surrounding (j) units, of a given view. It is clear that there are as many object units as there are possible views of the object, which in our case, is equivalent to the number of features in the object.

4. IMPLEMENTATION

Ideally, the system would utilize an array of feature detectors that completely cover the input image and process information in parallel. Similarly, the system would benefit from a large number of feature classes, since they would provide richer and more detailed description of objects. However, in order to make a system run in real time on a regular computer and without dedicated preprocessing hardware, we had to make several approximations.

Feature Selection. In our current implementation, we represent a car as a collection of only horizontal and vertical edges. Since an edge is an extended spatial object, we choose to specify its location in terms of the location of its central point. In this way, a car is modeled as a collection of points, arranged in 3D space, where each point represents an edge of specific size and

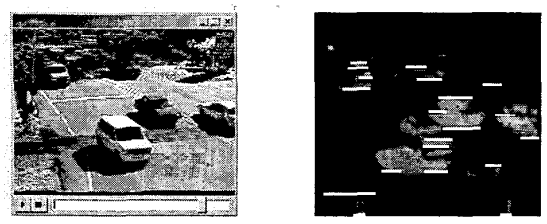

Figure 2: Original image (left) and processed image (right) that illustrates some of the prominent horizontal edges.

orientation. Using the statistics for the car sizes and their edges, one can easily calculate the mean size μ_j and the variance ν_j for each edge. Given the location of the fixation point and knowing the variations in size for every edge, it is then straightforward to propagate these uncertainties and calculate the regions where the centers of the edges should be. In order to map this 3D configuration of regions into a 2D image plane we use perspective transformation equations as described in [1]. In this way, for a given location of the fixation point within an image, we associate a group of 2D regions for allowable locations of the edge centers. Each such region represents a receptive field of one simple unit of the network.

Feature Detectors. Another approximation is related to the construction and use of edge detectors. Instead of having an array of edge detectors for detecting the horizontal and vertical edges of all the sizes, the system extracts only the prominent edges (with activations above the predefined threshold) and estimates their sizes. This task is accomplished by the preprocessing module. A detailed description of the preprocessing algorithm is outside of the scope of this paper and will be described in more detail elsewhere. Figure 2 illustrates some of the edges extracted by the preprocessing module and their estimated sizes. In our current implementation, the preprocessing module operates on the difference image that is obtained as a difference between the original (gray-scale) image and the background image that contains no vehicles.

The value of the pixel (i, j) of the background image at time $t + 1$ is calculated using the updating rule

$$B_{t+1}(i,j) = B_t(i,j) + \alpha \cdot D_t(i,j) \cdot O(i,j), \quad (5)$$

where α is an updating constant (how often to update the background image), $D_t(i,j)$ is the difference between the pixel values at times $t + 1$ and t and $O(i,j)$ is 1 if the pixel (i,j) belongs to an object that has been identified and 0 if it is part of the background. Therefore, the current image is used for updating the background image after the object identification is performed on the current image.

Each edge detector is selective to only an edge of a specific orientation, but can detect edges of various sizes around the preferred size. Since the distribution of sizes for any given car edge is fairly uniform, we use a Gaussian distribution to model the probability of an edge having a specific size. Therefore, an edge detector for an edge of horizontal/vertical orientation is specified with two parameters: the mean length of an edge and its variance. The input to the edge detector (of a given orientation) is an edge of specific size l and the output is the probability that measures how well this edge matches the expected edge size, $d = const * exp(-(\mu - l)^2/\nu^2)$.

5. RECOGNITION PROCESS

The recognition process starts with selection of the most prominent edge in an image, the one with the highest activation. The center of this edge becomes the fixation point from which the locations of other edges are measured. The system now has to determine whether the central edge represents an edge of a vehicle, and if it does which edge it represents. This is done by positioning all of the object units over the fixation point and measuring how much they are activated by the arrangement of the edges surrounding the central edge. The object unit with the highest activation selects some of the neighboring edges as representing a car and the central edge is given an identity as being a specific edge of a car (e.g. the bottom horizontal edge). In order to associate the group of edges (the central edge and the surrounding edges) as a car as opposed to noise, the activation of the object unit has to be above some predefined threshold. Once the group of

edges is selected as a representative of a car, their activations are suppressed and the system makes a saccade on another prominent edge and the previous procedure is repeated. The system makes as many saccades as there are prominent edges.

6. SUMMARY AND RESULTS

In this work we presented a biologically inspired system for car identification from video streams. The architecture of the network reflects the properties of foveal vision through the arrangement and sizes of the simple units. During the recognition process, the system explores the input image in a way similar to human saccadic movements, probing and analyzing different locations of the input at different times. The computational complexity associated with searching the space of edge activations is greatly reduced using selective attention thus allowing the system to process information in real time. The architecture described in this paper is implemented on a Pentium III, 700MHz processor using an input from a simple web camera.

We tested the performance of the system on several thousand video sequences. Once a system detects a still car it locks onto it (although it might fixate on different edges at different times) and the recognition is almost 100%. If the cars are moving and are separate from one another, the recognition accuracy is around 90%. However, when the cars become close to one another the recognition drops to about 70%, depending on how close the cars are and how much they are occluding each other. The system mistakenly recognized a van as a car with about 30% confidence. It never substituted a pedestrian for a vehicle and locked onto side road clutter in less that 1% of the time.

The system's performance regarding the correct identification of cars does not deteriorate if the preprocessing module extracts edges from a gray-scale image as opposed to a difference image. However, in that case, the number of false alarms is higher. Most of the false alarms are located on the sides of the road (the regions that contain significant edge-like structures) and can easily be filtered out using the road model.

The fact that we use feature-based and feature-centered object representation allows translation invariant recognition and makes the system very robust to occlusions. Similarly, the system can easily deal with variable lighting conditions since the features are edges and their extraction is not affected with overall change in illumination. One of the consequences of edge-based object representation is that the system can detect both still and moving cars equally well. We believe that the system's performance will be further improved with the inclusion of a richer set of features (in addition to only horizontal and vertical edges) and with a larger number of object classes.

7. REFERENCES

[1] Rafael Gonzales and Richar Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, 1993.

[2] S-W. Lee, H. H. Bulthoff and T. Poggio (Eds.), Biologically motivated computer vision. Berlin: Springer-Verlag, 2000.

[3] D. Koller, J. Weber, J. Malik, Robust multiple car tracking with occlusion reasoning, *Proceedings 5th European Conference on Computer Vision*, Springer-Verlag, Berlin, pp. 189-196, 1994.

[4] A. Lipton, H. Fujiyoshi, F. Patil, Moving target classification and tracking from real-time video, *WACV*, Princeton, 1998.

[5] J. Keller, S. Rogers, M Kabrisky and M. Oxley, Object Recognition Based on Human Saccadic Behaviour, *Pattern Analysis and Applications*, Vol. 2, Springer-Verlag, London, pp. 251-263, 1999.

[6] D. Koller, K. Danilidis, H. Nagel, Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes, *International Jouran of Computer Vision*, 10-3, pp. 257-281, 1993.

[7] P. Neskovic and L. Cooper. Neural network-based context driven recognition of on-line cursive script. In *IWFHR-7*, pp. 352–362, 2000.

[8] P. Neskovic, P. Davis, and L. Cooper. Interactive parts model: an application to recognition of on-line cursive script. *NIPS*, pp. 974–980, 2000.