Object Recognition by a Cascade Of Edge Probes

Owen Carmichael and Martial Hebert
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, USA
{owenc,hebert}@ri.cmu.edu

Abstract

We frame the problem of object recognition from edge cues in terms of determining whether individual edge pixels belong to the target object or to clutter, based on the configuration of edges in their vicinity. A classifier solves this problem by computing sparse, localized edge features at image locations determined at training time. In order to save computation and solve the aperture problem, we apply a cascade of these classifiers to the image, each of which computes edge features over larger image regions than its predecessors. Experiments apply this approach to the recognition of real objects with holes and wiry components in cluttered scenes under arbitrary out-of-image-plane rotation. [1]

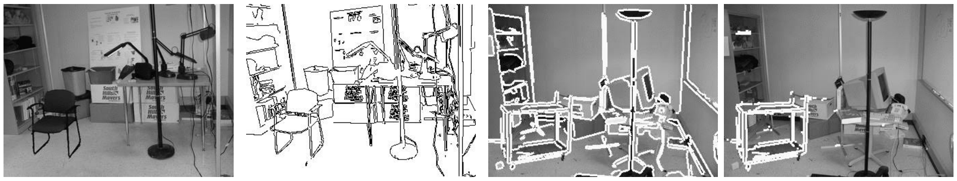
1 Introduction

Over the past 10 years, significant progress has been made toward the recognition of real, complex objects in cluttered scenes. There are now object recognition systems whose detection and false alarm rates are encouraging for real-world applications[29]; recently a real-time detector with comparable performance has even emerged[32]. The most common target object searched for is the human face, but in principle these systems could be trained to detect any of a variety of objects including cars and buildings.

Many of these approaches formalize the recognition problem as one of modeling the appearance of a rectangular image patch circumscribing the object, across changes in pose[26], lighting[4], or other conditions. Thus, the recognition problem is reduced to examining a specific rectangular image template, and using its appearance to decide whether or not it is the bounding box around the image of the target object.

Since the problem is formulated in terms of rectangular image windows, appearance-based recognition methods tend to work well when applied to target objects whose projection into the image fills a rectangular region. However, many objects produce images that are poorly approximated by rectangles; for objects such as the chair, table, and lamps in Figure 1, their bounding boxes in the image will contain a high percentage of pixels which map to the background or other objects. Most successful recognition techniques can handle the variation in template appearance induced by a small number of background

[1] This research was supported in part by NSF Grant IIS-9907142.



(a) Example test image

(b) Extracted edges

(c) Test image with edges overlaid

(d) Recognition result

Figure 1: We address the recognition of objects like chairs(1(a)) and carts(1(c)) based on edge cues (1(b)). An example result for the cart is shown in 1(d). See Section 1 for an overview and Section 4 for details on experiments.

pixels in the patch. When most of the template consists of clutter, however, its appearance can vary widely due to a modification of the background or object pose, making it difficult to detect the object based on the entire template.

A possible solution to this problem, proposed by several authors[6][11][22][30], is to break up the image representation of the object into a collection of smaller rectangles, each of which corresponds to a sub-section of the object. This strategy may be effective for some objects; consider, however, the recognition of an object containing elongated, wiry components such as the chair in Figure 1(a). Any image template larger than a few pixels across will intersect mainly clutter pixels when placed over any portion of the legs or armrests, and it is doubtful that image patches a few pixels square will contain sufficient information to discriminate the appearance of the object from the background.

Furthermore, popular approaches to object recognition analyze the greylevel or color texture patterns in candidate patches; thus they tend to work well when the target object has significant visual texture. Faces, cars, and buildings tend to possess this characteristic. But for the objects in Figure 1, along with many other common objects, there will be too little appearance variation to use texture as a cue for discrimination. Thus, while template-based techniques are effective for some objects, we feel it is worth investigating the problem of recognition from alternative cues, especially shape. Specifically, we hope to use machine learning techniques to boost the effectiveness of the contour-based recognition paradigm popular in the 1980s[16] to the point of feasibility in high-clutter scenes under significant 3D pose variation.

In this paper we address the problem of using shape cues to detect a particular object, such as the specific chair in Figure 1(a), across varying poses. Specifically, given an input image $\mathcal{I}$ (Figure 1(a)), we extract binary edges (Figure 1(b)) and use the configuration of the edges to determine which edge pixels belong to an instance of a target object, and which edge pixels belong to clutter (Figure 1(d)). Formally, let L denote a list of the pixels $q = [x, y]$ such that an edge has been detected at $\mathcal{I}(q)$. Our goal is to use L to recover a second list, O, which contains only those edge pixels $q \in L$ which correspond to points on our target object. Our only source of training data is a set of images containing the target object in typical scenes, from which edges have been extracted and labeled “object” or “clutter.” In other words, at training time we are given a set of images $\mathcal{T} = \{\mathcal{T}_1, \cdots \mathcal{T}_{n_t}\}$ and a set of edge lists, $T = \{T_1, T_2, \cdots T_{n_t}\}$, where each T_j is composed of two sub-lists T_{j+} and T_{j-}; T_{j+} consists of edge pixels q_+ extracted from image $\mathcal{T}_j$ which correspond to a point on the target object, while each $q_{j-} \in T_{j-}$ is an edge pixel which maps onto the background. Given the edge list O, alignment techniques may be applied for verification purposes or to estimate object pose[3][31]; also, it is possible to summarize the location of the object in the image by computing a bounding box and centroid from O. Here,

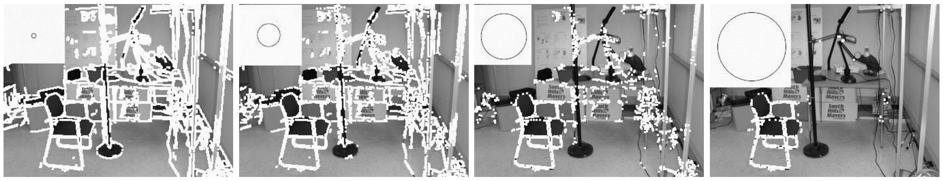

(a) Phase 1 (b) Phase 5 (c) Phase 10 (d) Phase 15

Figure 2: Example recognition results at successive phases of the recognition cascade. The size of the aperture for each phase is depicted by the circle at upper left. Edge points classified as “chair” are shown in white. See Section 3.2 for an overview and Section 4 for details on experiments.

however, we focus on the problem of object-background separation in edge images, *i.e.* determining which edge pixels correspond to the object and which correspond to clutter.

We present a cascade approach to recovering O. At each point on each edge, we examine the edges in a neighborhood surrounding it, which we call the *aperture* of the edge point(Figure 3(a)). How edges are arranged inside the aperture, termed the *local edge configuration* of the edge point, is the cue used to determine whether that edge pixel belongs to the object or the background. A classifier is trained from the example views to discriminate local edge configurations of clutter edge points from those of object edge points. Unfortunately, if the aperture is too small, the local edge configuration may be ambiguous; in other words, it might be impossible to tell which class the edge point belongs to based on edge information inside the aperture. For this reason, ambiguous edge points are passed on to a second classification phase, which considers the local edge configuration in a larger aperture. If it is still unclear at this stage whether the edge point belongs to the foreground or background, we attempt to classify it based on features in a larger aperture, and so on. As an illustration, Figure 2 depicts four phases in this process for the recognition of the chair in the lower left portion of the image.

At each phase in the cascade, a discriminative classifier computes a sparse set of localized edge features which measure edge density in some image neighborhood. The locations of the edge features are determined according to a tree structure which is learned at training time. Figure 3(c) illustrates the classification of one edge point, at one phase of the cascade.

2 Related Work

Object recognition research in the 1980s culminated in systems which could detect occluded, 2D, non-convex shapes from binary edge images[16]. Interpretation trees[16], for example, use a tree search to explore the space of all possible correspondences between features on an object model and features in the image. Unfortunately, as the number of model features and image features grows, the space of correspondences can grow intractably large, especially if the image contains significant clutter or noise.

Indexing techniques such as geometric hashing[23] suffer in the presence of clutter as well. In these approaches, each *k*-tuple of image features casts votes for the identities and/or poses of objects in the image, based on their geometric arrangement. If the image contains significant noise[17] or clutter, the votes cast by sets of clutter features will overwhelm the votes cast by the object, making it difficult to draw any conclusions about what objects are there.

More recently, Belongie *et al*[5] extended the notion of 3D shape signatures[18] to

2D shape, for the purpose of edge-based recognition. At each edge point in an image, a histogram, or "shape context," is calculated; each bin in the histogram counts the number of edge pixels in a neighborhood near the point. Nearest-neighbor search then determines correspondences between shape contexts from a test image and shape contexts from model images. Our approach is closely related; both use the distribution of edges in an aperture surrounding a point as the fundamental feature for recognition. However, the shape context uses a “dense” set of edge features for recognition; in other words, the bins in the histogram exhaustively cover the entire aperture. Since we only compute edge features at isolated image locations deemed likely to discriminate the edge point in question as object or clutter, the features we use are spatially “sparse.” While dense features may be effective when the background is not a concern, we feel that they will represent local edge configurations poorly for the target objects and scenes we consider. Specifically, if the neighborhood surrounding an object edge contains a significant number of background edges– consider the cart in Figure 1(c)– then many of the shape context bins will be filled solely with background edge points. This is largely the same reason why rectangular templates represent local appearance poorly on wiry objects: much of the local object representation will actually consist of image data drawn from clutter.

Other researchers have addressed the problem of recognizing objects by finding k-tuples of specific appearance features arranged in appropriate ways[10][2]. These techniques rely on checking the configurations of every k-tuple of detected features in an image neighborhood; as the number of features in the configuration grows, and the density of candidate features grows, there will be a combinatorial growth in the number of scores to be given out at run time.

In [10], a joint Gaussian model of feature locations is assumed; similarly, other recognition approaches assume that the distribution of object features in an image can be described by a Markov random field [24][7][25] or an object-specific model such as a body plan[14]. Our objects are distorted by arbitrary out-of-image-plane rotation; Gaussian, Markovian, or other simplified models may not capture the variation in feature configuration induced by these transformations.

3 Approach

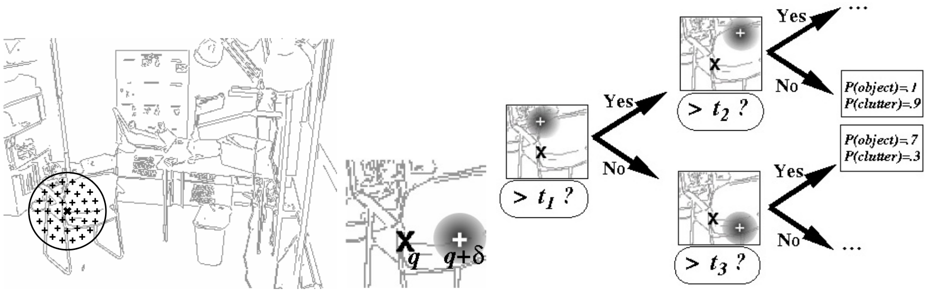
3.1 Edge Probes

We begin by defining the edge features our algorithm will use to describe the local edge configuration in an image region. An *edge probe* at *probe center* p over a list of edge pixels L is defined as

$$EP(p, L) = \sum_{t \in L} \exp\left(-\frac{\|p - t\|^2}{\sigma^2}\right)$$

where t and p are 2-vectors of $[x, y]$ image coordinates. An edge probe can be thought of as a Gaussian receptive field with variance σ^2, centered at point p in an edge image whose edge pixels are contained in the list L. Edge probes measure the density of edge pixels in some neighborhood in the image; in this sense, each edge probe is analogous to a bin in a shape context histogram[5].

Our goal is to determine, for each *query edge pixel* $q = [x_q, y_q]$ extracted from an image, whether it belongs to an instance of our target object or whether it was produced by the background. We will use edge probes computed at probe centers in an aperture



(a) Aperture and shifted probe centers (b) An edge probe (c) Classification of a query edge point

Figure 3: Overview of one phase of the recognition cascade. 3(a) Edge probes are evaluated in an aperture surrounding a query edge point. The query edge point is marked "X," and edge probes are evaluated at locations marked "+."3(b) Each edge probe measures edge density in some image neighborhood. Here an edge probe is evaluated at shifted probe center $q + \delta$ for a query edge point q. 3(c) Edge points are classified by evaluating edge probes according to a tree structure. See Sections 3.1 and 3.3 for details.

surrounding q to make this decision. Specifically, consider a set of *relative probe centers* $\Delta = \{\delta_1, \delta_2, \cdots, \delta_k\}$, $\delta_i = [x_{\delta_i}, y_{\delta_i}]$, laid out over a 2D grid centered at the origin. To classify q, we shift the relative probe centers so that they surround q and compute a subset of edge probes $EP(q, \Delta, L) = \{EP(q + \delta_1, L), \cdots EP(q + \delta_k, L)\}$ at *shifted probe centers* $\{q + \delta_1, \cdots, q + \delta_k\}$.An illustration is shown in Figure 3(a)-3(b). We emphasize that there is a critical distinction between the aperture of an edge point (denoted by the black circle in Figure 3(a)) and the spatial support of a single edge probe (denoted by the gradient-shaded region surrounding $q + \delta$ in Figure 3(b))– the aperture describes the image region over which *all* edge features for a given query point are evaluated, while the portion of the image which contributes to a *single* edge feature is determined by the edge probe support.

Given a fixed σ, we space the relative probe centers evenly over a circular aperture as in Figure 3(a) so that each pixel in the aperture contributes to one or more edge probes. But how large should the aperture be? We want the shifted probe centers to cover a large enough neighborhood surrounding q that the edge probes will contain sufficient information to discriminate object pixels from background pixels. At the same time, however, if the aperture is too large (covering the entire image, for example), an unfeasible amount of computation will be required at training time to evaluate edge probes that might not be crucial for classification. Worse, if the aperture is so large that most of the edge probes at shifted probe centers are totally irrelevant to the category of the query edge point, error-prone classifiers could be trained[20][1]. Thus, we are presented with "the aperture problem" which appears in so many computer vision problems– when attempting to induce information about a particular location in the image we want to incorporate image data from a large enough surrounding area that the information can be induced, but not so large that we introduce irrelevant data or useless computation.

3.2 The Cascade

Consider a set of relative probe centers Δ which cover a circular aperture as in Figure 3(a). Define $A(\Delta)$ to be the radius of the circle. Our approach is to train a series of classifiers $f_1, f_2, \cdots, f_k$ which evaluate edge probes according to sets of relative probe centers $\Delta_1, \Delta_2, \cdots, \Delta_k$ such that $A(\Delta_1) < A(\Delta_2) < \cdots < A(\Delta_k)$. The first classifier in the series, f_1, is trained to classify edge points based on edge probes taken from a small radius surrounding them; f_2 classifies based on edge probes over a slightly larger radius, and so on. Edge points labeled "object" by $f_1$are classified by f_2; points labeled "background" by $f_1$are discarded. Edge points labeled "object" by f_2 are passed to f_3, and so on.

Thus, we solve our aperture problem in phases– we first identify those edge points whose class is discriminable based on very nearby features, then identify points that are made discriminable by features slightly farther away, and continue to do so until the aperture covers the entire object in question. As an illustration, Figure 2 shows the results of classification at four phases of the cascade.

Besides providing a solution to our aperture problem, the classifier cascade allows fast screening of image locations that are easily discriminable from the object of interest based on information in a small window, leaving the bulk of the computation to more ambiguous sections of the image. Similar cascade strategies have recently achieved significant speedups for template-based approaches to recognition[32][19].

3.3 Classifiers

We seek classifiers $f_1, f_2, \cdots, f_k$ which compute a sparse set of edge probes over apertures $\Delta_1, \Delta_2, \cdots \Delta_k$. To accomplish this, we design each classifier as a decision tree of edge probes[28]. A classifier f consists of a set of nodes connected in a tree structure; each node represents the evaluation of an edge probe at some shifted probe center. To classify a query edge point $q \in L$, we start at the root of the tree and evaluate $EP(q + \delta_1, L)$, where δ_1 is the relative probe center associated with the root of the tree. Depending on whether the value of the edge probe is greater or less than some threshold t_1, we shift to one or the other of the children of the root node, and evaluate edge probe $EP(q + \delta_2, L)$ where relative probe center δ_2 is associated with the child. This process of moving from node to node based on the evaluation of edge probes continues until a leaf node is encountered. Associated with each leaf node is a table which gives the probability that the query edge point belongs to the object. By setting a threshold on this probability, we arrive at a binary decision about whether the image point is classified as an object point or background point. In short, the application of a classifier to an edge point consists of a series of edge probes whose probe centers are dependent on the structure of the tree. Figure 3(c) illustrates the application of one classifier to one query edge point.

Our training procedure for the decision trees is a two-step process of tree generation and pruning, following the reduced-error pruning approach of Quinlan [28]. In this framework, the training data is split into two subsets, which we will refer to as the *tree-growing set* and the *holdout set*. Standard tree induction techniques are used to build a decision tree with high classification accuracy on the tree-growing set; then, subtrees are pruned from the tree when doing so improves some performance criterion on the holdout set[28][8][9]. Our pruning criterion is shaped by the fact that the classifiers are applied in a cascade. Specifically, consider an edge pixel q which corresponds to a point on the object. If a classifier mistakenly classifies q as clutter (we will refer to this as a "false negative"), then the edge point is permanently removed from consideration by further classifiers in the cascade; however, if a clutter edge pixel q is mistakenly classified as belonging to the object (a "false positive"), then the edge point is passed on to later phases in the cascade, which may in turn re-classify it correctly based on edge information in a larger aperture. For this reason, we optimize a Neyman-Pearson criterion [12] during pruning; specifically, we prune subtrees whenever doing so improves the false positive rate of the classifier while keeping the false negative rate below a low, fixed threshold θ.

The training and run-time behavior of our algorithm are summarized in Algorithm 1 and Algorithm 2. Here, for a given set of edge lists $L = \{L_1, \cdots L_{n_l}\}$, $f(L) = \{f(L_1), \cdots f(L_{n_l})\}$, and $f(L_j) = \{l_{jq} \in L_j | f$ classifies l_{jq} as 'object'$\}$. L_{j+} is the sub-list of L_j containing object edge pixels extracted from corresponding image and L_{j-} contains clutter edge pixels.

Algorithm 1 Training

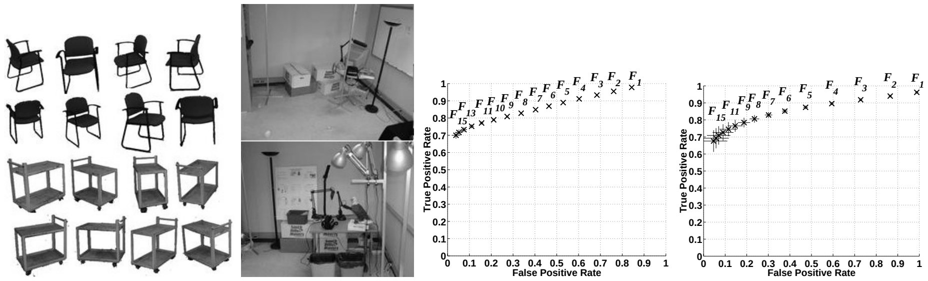
```
Require: Edge lists $T = \{T_1, T_2, \cdots T_{n_t}\}$, sets of probe centers $\Delta = \{\Delta_1, \cdots \Delta_{n_d}\}$, $\sigma$, $\theta$.
Split $T$ into a tree-growing set $G = \{G_1, \cdots G_{n_g}\}$ and a holdout set $H = \{H_1, \cdots H_{n_h}\}$.
$G^1 = \{G^1_1, G^1_2, \cdots G^1_{n_g}\} = G$, $H^1 = \{H^1_1, H^1_2, \cdots H^1_{n_h}\} = H$
for $i = 1$ to $k$ do    // loop over cascade phases
    for all $G^i_j \in G^i$ do    // loop over the tree-growing set
        for all $q_+ \in G^i_{j+}$ do    // loop over object edge pixels
            $g^i_{jq+} = EP(q_+, \Delta_i, G^i_j)$
        end for
        for all $q_- \in G^i_{j-}$ do    // loop over clutter edge pixels
            $g^i_{jq-} = EP(q_-, \Delta_i, G^i_j)$
        end for
    end for
    Train a decision tree $f_i$ to discriminate $\{g^i_{1q+}, \cdots g^i_{n_g q+}\}$ from $\{g^i_{1q-}, \cdots g^i_{n_g q-}\}$.
    Prune $f_i$ based on $H^i$.
    $G^{i+1} = f_i(G^i)$, $H^{i+i} = f_i(H^i)$    // discard correctly classified background edge pixels
end for
```

Algorithm 2 Run Time

```
Require: List of edge pixels $L$
$L^1 = L$
for $i = 1$ to $k$ do    // loop over cascade levels
    $L^{i+1} = f_i(L^i)$    // discard background edge pixels
end for
Return $L^{k+1}$
```

4 Experiments

To validate our approach we address the problem of detecting two common objects, a chair and a cart, in highly cluttered indoor scenes under high variation of out-of-image-plane rotation. We evaluate the performance of our cascade by computing the true positive (*i.e.* one minus false negative) rate and false positive rate in each image. We emphasize that since we represent objects at a pixel level, "true positive rate" does not mean "percentage of times the object was detected." Instead, it means "percentage of object *pixels* detected." Thus, even if the true positive rate is below 100%, it may still be possible to conclusively locate the object in all test images, since the density of object edge points will be relatively high at the true location of the object if the true positive rate is relatively high. Likewise, "false positive rate" does not relate to "number of times a section of the background was mistakenly labeled as the object," but rather to "number of times an *edge pixel* in the background was mistakenly labeled as belonging to the object." Thus, even if the false positive rate is greater than zero, it may be possible to achieve zero false detections of the object, especially if the falsely-detected background edge pixels are sparsely distributed in the scene.



(a) Foreground images (b) Background images (c) ROC curve for the chair (d) ROC curve for the cart

Figure 4: Example foreground and background images and ROC curves for recognition experiments. See Section 4.

4.1 Procedure

We took 150 images each of the cart and chair against a blue screen (Figure 4(a)). The images span the full revolution of the objects in the plane parallel to the floor. The elevation of the camera varies by approximately 25 degrees with respect to the object, and the extent of scale variation across images is about 10%. For some of our images, the generic viewpoint assumption is violated[15]; for example two of the legs of the chair in Figure 4(a), second row, fourth column, are accidentally aligned.

We also took images of a background scene consisting of a set of "office" objects– for example lamps, a table, and boxes(Figure 4(b)). The set of views spans roughly 60 degrees of rotation in the plane parallel to the floor, and variation in scale and camera elevation is about the same as for the cart and chair images. To induce appearance variation in the background between views, we modified the poses of each background object and shuffled their relative positions every 5 to 8 images. The camera was moved between each view.

The images used as training and testing data in our experiments are composites of random pairs of foreground and background images (Figure 1(a) and Figure 1(c) are examples). The pairing was done without replacement; in other words, there is no repetition of foreground or background images in the composite images.

For each recognition trial, 150 composite images were partitioned into a tree-growing set of 50 images, a holdout set of 50 images, and a test set of 40-50 images. Edges were detected on all images using the Vista line finder[27]; for computational reasons we sample the detected edges at 5 pixel intervals and classify the edge samples. In each image, the ratio of the number of background edge pixels to foreground edge pixels is approximately $10 : 1$.

We arranged the probe centers in concentric rings as in Figure 3(a). The rings are spaced at radial intervals of σ pixels, where σ^2 is the edge probe variance. The set of relative probe centers Δ_i used for training f_i is the set of all relative probe centers within a distance r_i of the origin, where $r_i = \sigma i$. There are 15 classifiers in the cascade, so the apertures for the classifiers vary in radius from σ to 15σ pixels. We conducted three distinct sets of recognition trials, with σ set to 5 pixels, 10 pixels, and 20 pixels respectively; we achieved similar recognition performance regardless of the setting of σ, so we only report results for $\sigma = 10$ pixels here.

To train the classifiers, we first discretize the edge probe values using the implementation of minimum-entropy discretization[13] in the MLC++ software library[21]; decision trees were grown using the ID3 routine from the same package. The Laplace approximation was employed to estimate probabilities at each leaf in the tree. We used the holdout set to prune the decision trees as described in Section 3.3, setting our target false negative rate θ to 2%.

4.2 Results

A sample result on the chair is shown in Figure 2; the images show classification results after the first, fifth, tenth, and fifteenth cascade phase. Note that as successive classifiers are applied, using larger apertures, the number of background edge points is reduced dramatically while retaining a high number of edge points on the chair. Note that the false positives are so sparse and isolated after the last phase that they could easily be removed by simple filtering. A sample input and result for the cart object are shown in Figure 1(c) and Figure 1(d). Again, most of the background pixels have been filtered out by the cascade, while a high concentration of edge pixels remains on the cart.

The performance of each tree in the cascade, over all test images containing the chair, for $\sigma = 10$ pixels, is summarized in the ROC curve in Figure 4(c). We performed 7 recognition trials; each trial consisted of randomly partitioning the images into tree-growing, holdout, and test sets, training the cascade, and evaluating the overall false negative and false positive rates of the cascade as more phases are added. Thus, the point marked F_1 plots true positive rate as a function of false positive rate for a cascade consisting of one classifier, f_1; F_5 plots the performance of a cascade containing $f_1, f_2, \cdots, f_5$; and so on. More specifically, for each cascade we compute the true positive rates $N = \{n_{ij}\}$ and false positive rates $P = \{p_{ij}\}$ for each test image i and recognition trial j. The “x” in the graph plots $(mean(P), mean(N))$; error bars extend to the left and right by $var(P)$ and up and down by $var(N)$. Figure 4(d) shows an analogous graph for results of 6 recognition trials with the cart. For both objects, the results for $\sigma = 5$ pixels and $\sigma = 20$ pixels are similar.

The true positive and false positive rates for the two objects are comparable– for example, roughly 70% of edge pixels on the object are retained, versus 5% false positives among the background.

5 Conclusion

Our approach to separating objects from background based on edge cues consists of screening each edge pixel in the image through a series of classifiers, each of which computes edge features over successively larger image areas. Each classifier in the cascade computes a sparse set of localized edge features in a sequence determined by its tree structure. By tuning feature extraction to the object and background present in training images, we overcome the effects of object structure (concavities, holes, wiry structures) which tend to confound template-based approaches to recognition. And by screening the image through a series of increasingly complex classifiers, we quickly discard edge points which are easily discriminated from the object, saving computation for more ambiguous portions of the image.

References

[1] Hussein Almuallim and Thomas G. Dietterich. Learning with many irrelevant features. In *Proc. AAAI*, 1991.

[2] Yali Amit, Donald Geman, and Bruno Jedynak. Efficient focusing and face detection. Technical Report 459, Department Of Statistics, University of Chicago, October 1997.

[3] Ronen Basri and David Jacobs. Projective alignment with regions. *IEEE Trans. PAMI*, 23(5):519–527, May 2001.

[4] P. Belhumeur and D. Kriegman. What is the set of images of an object under all possible illumination conditions? *IJCV*, 28(3):245–260, 1998.

[5] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI*, 24(4):509–522, April 2002.

[6] J. Ben-Arie, Z. Wang, and R. Rao. Iconic recognition with affine-invariant spectral signatures. In *Proc. ICPR*, volume 1, pages 672–676, 1996.

[7] Yuri Boykov and Daniel P. Huttenlocher. A new bayesian framework for object recognition. In *Proc. CVPR*, pages 517–523, 1999.

[8] Jeffrey P. Bradford, Clayton Kunz, Ron Kohavi, Cliff Brunk, and Carla E. Brodley. Pruning decision trees with misclassification costs. In *Proc. ECML*, pages 131–136, April 1998.

[9] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth International, Belmont, CA, 1984.

[10] M. Burl, M. Weber, and P. Perona. A probablistic approach to object recognition using local photometry and global geometry. In *Proc. ECCV*, pages 628–641, 1998.

[11] Vincent Colin de Verdiere and James L. Crowley. Visual recognition using local appearance. In *Proc. ECCV*, pages 640–654, 1998.

[12] Richard Duda, Peter Hart, and David Stork. *Pattern Classification*. Wiley-Interscience, 2 edition, 2001.

[13] U.M. Fayyad and K.B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. IJCAI*, pages 800–805, 1989.

[14] D.A. Forsyth and M.M. Fleck. Body plans. In *Proc. CVPR*, pages 678–683, 1997.

[15] W.T. Freeman. Exploiting the generic viewpoint assumption. *IJCV*, 20(3):243–261, 1996.

[16] W.E.L. Grimson. *Object recognition by computer : the role of geometric constraints*. MIT Press, 1990.

[17] W.E.L. Grimson and D. Huttenlocher. On the sensitivity of geometric hashing. In *Proc. ICCV*, pages 334–338, 1990.

[18] Andrew Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. PAMI*, 21(5), 1999.

[19] Daniel Karen, Margarita Osadchy, and Craig Gotsman. Antifaces: A novel, fast method for image detection. *IEEE Trans. PAMI*, 23(7):747–781, July 2001.

[20] K. Kira and L. A. Rendell. A practical approach to feature selection. In *Proc. ICML*, pages 249–256, 1992.

[21] Ron Kohavi, Dan Sommerfield, and James Dougherty. Data mining using MLC++: A machine learning library in C++. In *Tools with Artificial Intelligence*. IEEE Computer Society Press, 1996. `http://www.sgi.com/tech/mlc`.

[22] John Krumm. Object detection with vector quantized binary features. In *Proc. CVPR*, pages 179–185, June 1997.

[23] Y. Lambdan and H. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proc. ICCV*, pages 238–249, 1988.

[24] S. Z. Li and J. Hornegger. A two-stage probabilistic approach for object recognition. In *Proc. ECCV*, 1998.

[25] John MacCormick and Andrew Blake. Spatial dependence in the observation of visual contours. In *Proc. ECCV*, pages 765–781, 1998.

[26] H. Murase and Shree Nayar. Visual learning and recognition of 3-d objects from appearance. *IJCV*, 14:5–24, 1995.

[27] Arthur Pope and David Lowe. Vista: A software environment for computer vision research. In *Proc. CVPR*, 1994.

[28] J.R. Quinlan. *C4.5 : programs for machine learning*. Morgan Kaufmann Publishers, 1993.

[29] H. Schneiderman and T. Kanade. Probabilistic modeling of local appearance and spatial relationships for object recognition. In *Proc. CVPR*, 1998.

[30] A. Selinger and R. C. Nelson. A perceptual grouping hierarchy for appearance-based 3d object recognition. Technical Report 690, University of Rochester Computer Science Department, May 1998.

[31] Paul Viola and William M. Wells III. Alignment by maximization of mutual information. *IJCV*, 24(2):137–154, 1997.

[32] Paul Viola and Michael Jones. Robust real-time object detection. Technical Report CRL 2001/01, Compaq Cambridge Research Laboratory, 2001.