ABSTRACT
With the growing image collection on the web, classifying images has become an actively explored problem. In this paper we present a novel approach to the classification of images depicting objects in a category using the odd-man-out principle of visual categorization. Specifically, we build a model of an object category by noting discriminative features that are commonly observed across the member images of the class. Appearance changes due to pose, illumination and intra-class variations are modeled using multi-scale affine kernels. The best matching affine kernel for a given query image is found as the one that has the largest overlap of discriminable features that are commonly observed across the class. We show that using the odd-man-out principle of IQ tests not only results in better feature selection but also in more robust object class categorization, in comparison to the state-of-the-art methods on large benchmark image datasets.

Categories and Subject Descriptors
D.3.3 [Algorithms and Methods]: Computer Vision, Image categorization.

General Terms
Object categorization, image retrieval, pattern recognition, affine transformation, class categorization.

Keywords
Class categorization, affine kernels, image retrieval, image indexing, object recognition, object categorization.

1. INTRODUCTION
With the growing image collection on the web, classifying images has become an important problem. Popular search engines such as Google and Yahoo now offer text-guided image search. The quality of matches retrieved, however, depends heavily on the appropriateness of annotation attached to images, either from their file names or from the surrounding context/captions in documents/web pages. Recent work in automatic image categorization tries to improve the results of such searches by recognizing the semantic categories of scenes[8,10,14]. This is a challenging problem, in general.

Objects depicted in scenes appear in varying pose, illumination, and form against complex backgrounds. Figure 1 illustrates the different appearances objects can take that belong to the same semantic category. Figure 2 illustrates the further difficulty of this problem when the visual object category has to be assessed independent of the variety of backgrounds against which the objects can appear.
recognition? Given the vastly superior performance of human vision on this task, it is reasonable use it as an inspiration. The Intelligence Quotient (IQ) tests [19] have been traditionally testing these skills through exercises such as the ones shown in Figure 3, where a subject is asked to pick the odd-man-out, or its converse, that is, find objects that belongs to the same class. In this paper, we adopt odd-man-out principle of visual categorization to develop a novel algorithm for recognition of object categories in natural scene images.

Specifically, we automatically extract features in images that bring out the commonality in the class in a way that is invariant to natural intra-class variations and yet discriminative enough to spot the outliers. Specifically, we build a model of a class by noting features that are consistently observed across member images that depict object of a category.

The changes in appearance of objects belonging to a category are modeled using multi-scale affine kernels. Given a query image, we assign it to a class in which it is not an odd-man-out. That is, it possesses those features that are consistently observed among the class members which are also discriminatory among classes.

We show that using the odd-man-out principle results in not only better feature selection but also in more robust object class categorization in comparison to the state-of-the-art methods.

The rest of the paper discusses the method in detail and is organized as follows. In Section 2, we motivate the approach by presenting the odd-man-out principle. In Section 3, we review relevant work in object categorization. In Section 4, we present the multi-scale affine kernel and its use in class learning and category recognition. In Section 4, we present results of object categorization produced by our algorithm on benchmark datasets and compare with two other popular methods of category recognition.

Our paper makes several novel contributions. First, we take a fresh look at the object categorization problem using the analogy of IQ tests to automatically extract common features across members of a class. Next, the affine kernel representation proposed is robust to illumination changes, pose changes and intra-class variations. Finally, the multi-scale affine kernel is used both in feature selection and nonlinear classification, avoiding the need for a separate linear classifier such as SVM that is used in existing approaches.

2. THE ODD-MAN-OUT PRINCIPLE

The ability to visually categorize objects develops early in humans. The Intelligence Quotient (IQ) tests [19] have been traditionally testing these skills through exercises where a subject is asked to pick the odd-man-out, or its converse, that is, find objects that belongs to the same class. Figure 3 shows examples taken from an actual test [19] administered to 4 year-olds that illustrates some of the important principles of visual categorization. Exercise D65 in Figure 3 illustrates a case in which the reversing of foreground/background colors in choice (b) makes this an odd-man-out. To select the correct answer, a subject looks at three ‘feature dimensions’, namely, color, orientation and shape. On an abstract shape axis, all choices have a common triangular shape and hence not separable by shape. On the orientation dimension, all choices seem to differ in orientation. Hence orientation is an irrelevant feature for categorization here. Finally, on the background/foreground color dimension, four of the members have the same background/foreground colors, while choice (b) seems to be the ‘odd-man-out’ in this dimension. A similar reasoning applies to case D-62 in Figure 3, where again the shape and orientation are common to all cases, but choice (c) differs from others in the merging of the darker regions and is hence the odd-man-out. Finally, cases D25 and D26 in Figure 3 show a case of the converse problem, where an object that belongs to the class represented by the objects on the left, has to be chosen. In case D26, both color and orientation are not relevant features for separation for different reasons. Choices a through d agree in color with the target class, hence color cannot be used to discriminate between them and the target class. On the other hand, orientation is different for all objects in the class on the left, hence not useful again for discrimination. Finally, the triangular shape is common to objects in the class and possessed by only one choice (b) on the right, making the rest odd-man-out choices. Lastly, case D25 shows a case where an even finer characteristics of shape is utilized in removing the odd-man-outs to select the correct choice of (c) based on concave pentagonal shape.

Figure 3. Illustration of visual category recognition in IQ testing.

From these IQ test exercises, we can draw some important principles for object categorization. We call them collectively as the Odd-man-out principle. According to this principle, a target object belongs to a category/class if these conditions are true:

1. Target object possesses features that are common to all members of the class. This is a necessary but not sufficient condition.
2. Features that are common to all members of the class are discriminatory only if they are not common to other members of other classes as well.
3. Target object can have arbitrary features in only those dimensions in which the members of the class themselves differ.

The odd-man-out principle can help in the problem of object category recognition in the following ways. First, it can help explain why members belong to a class. More importantly, it can be operationalized to develop a computational algorithm for automatic class recognition. Thus in the example categories
shown in Figure 1 row 1, the regularly spaced black and white bands of the accordion are the relevant common features to describe the accordion class. Similarly, corners at the intersection of the semi-circular arc with the vertical edge are discriminative common features observed in the anchor class. Finally, the blades of the fan are the discriminative common features observed across the fan class. If a method can automatically identify such common relevant features, the resulting category recognition will closely approximate the perceptual categorization of human vision, giving perceptually appealing reasons for the categorization.

3. RELATED WORK
The recognition of object categories is a well-researched problem in computer vision. Early work considered the problem of model-based recognition of the same object under affine and projective deformations [1,2]. These methods involved explicit search for correspondence features [1] or indexed search using variants of geometric hashing [2]. In contrast to model-based object recognition, object categorization has to not only deal with pose and illumination conditions, but also intra-class variability (e.g. different kinds of motorcycles) [8]. Early approaches to category recognition explicitly modeled changes between objects in a class using flexible shape models such as the constrained affine shape models [4], maximum likelihood models of shape and appearance [5], and contextual shape [6,7]. Scalable models for object class recognition for large image databases were developed by using the analogy with the bag of words models (e.g., probabilistic Latent Semantic Analysis (pLSA) [9] and Latent Dirichlet Allocation (LDA) [9]) from document retrieval. In these approaches, all object-containing images are represented by documenting the number of visual words (features with quantized values on their dimensions) occurring in the images in a co-occurrence matrix. The probabilities of a topic (an object category) occurring in an image based on the visual words (features) is estimated using the EM algorithm. To reduce the vocabulary size, clustering is often performed on the visual words to form a code book similar to vector quantization [10].

The spatial location information was ignored in early models [3] while later models incorporated position-specific information [10,8] by dividing the image on regular grids. The pyramid kernel variants [14,12] address spatial layout to a limited extent by partial histogram matching over a number of scales [14]. Instead of a bag of features, a multidimensional histogram is used in these methods. The distance between a pair of images is assessed using histogram intersection. The distance metric alone is not sufficient for class categorization. Further learning is performed using the histogram intersection distance over multiple scales using an SVM. Pairwise distances from a small set of randomly selected training images from manually categorized datasets are used to train standard classifiers such as Support Vector Machines (SVM)s, one per object category[14]. To provide robustness to translation and scale, the histogram kernel is applied in a scale-space fashion to form pyramid kernels [14]. However, matches at higher scale levels can be spurious. While weighting the higher scales less can alleviate some of the problem, it decreases the ability to tolerate rotational variations [14]. Capturing the complete spatial layout using affine-invariant descriptors would require explicit search for corresponding features, making it impractical in large image retrieval settings [12]. The work in [12] models spatial layout by aggregating statistics of local features over fixed sub-regions. A simple adaptation of the pyramid match kernels is done by separating the feature types and using the image as the registration grid [12]. Lastly, the work of Verma et al[16] explore combinations of kernels to achieve different levels of tradeoffs, and have achieved a benchmark performance on the Caltech-101 data set[15].

From the point of object category recognition, existing approaches suffer from some limitations. First, since the features are sampled on a regular grid and are not required to come from single objects, it limits their ability to extract commonly perceived features across the images in a class. Applying a grouping operation prior to feature selection can at least ensure that the feature collections map to spatially contiguous regions in images. Secondly, the representation of features does not adequately model the spatial layout seen in objects. Further they do not perform adequate feature selection to factor out the effect of background changes, pose changes as well as intra-class variations. Finally, when a class is characterized by a handful of common features scattered in the feature space, a separating hyperplane as assumed by SVM and other classifiers is not sufficient to group members of the class.

4. CATEGORY RECOGNITION USING AFFINE KERNELS
We now begin the description of our approach to category recognition using the odd-man-out principle. Applying the principle to objects in natural scenes presents several challenges. First, humans use several feature dimensions to analyze object categories. While the exercises in Figure 3 gave three (shape, color, orientation), in practice the number of such feature dimensions could be large, especially, when semantic categories have to be recognized. For example, inferring the category of vehicles from pictures of cars, motorcycles, and trucks with different visual appearances, requires using several ‘semantic’ feature dimensions. Secondly, even if we restrict to visually discriminable classes, considerable variations in object appearance under varying pose, illumination, and backgrounds can produce highly different appearances for objects belonging to the same visual class.

In our approach we work with a specific set of features and attributes that are robust to pose and illumination changes and model the spatial layout observed within object regions. As the focus is on the new category recognition methodology, we chose primarily geometric features to clearly illustrate the technique. Further augmentation with richer SIFT features with texture, color and gradients [18] would only improve the performance.

In what follows, we describe our choice of features, their use in building a class model, and in class recognition.

4.1 Choice of features
To ensure a choice of semantically meaningful features likely to come from single objects, we process images to extract curves. We then use curvature change points or corners along a curve as features. Working with curves formed from edges gives robustness to illumination changes. Figure 4 shows corner features along the curve C2. There are several readily available packages to extract curves and corner features from computer vision libraries [17].
Each feature is described by a collection of attributes. The attributes we select are in an affine-invariant coordinate system. That is, as shown in Figure 4, using 3 points on an object we set up a local coordinate system to describe the coordinates of all other points. Thus given three basis triple points \( B = < P_0, P_1, P_2 > \) where \( P_0 = (x_0, y_0) \), \( P_1 = (x_1, y_1) \), \( P_2 = (x_2, y_2) \) on an object, any new point \( O = (x, y) \) can be represented by affine coordinates \((\alpha, \beta)\) as

\[
(O - P_0) = \alpha(P_1 - P_0) + \beta(P_2 - P_0)
\]

It can be easily shown that \((\alpha, \beta)\) are affine-invariant. That is, when the object depicted in the image undergoes a pose change that can be modeled as an affine transformation \((A, T)\) modeling scale, rotation, stretch through the linear transform and translation \(T\), we have

\[
O' = A.O + T
\]

The coordinates with respect to the transformed basis points \((O', P_0', P_1', P_2')\) remain the same. Thus

\[
(O' - P_0') = \alpha(P_1' - P_0') + \beta(P_2' - P_0')
\]

Unless the camera is very close to the objects in a scene, images depicting the same object under a different pose can be easily modeled using such a coordinate system. The linear transform can also account for some intra-class variations within objects such as the differences in the chair instances shown in Figure 1.

Since the features are corners along a curve, more information is available to describe the corners than just their affine coordinates. In particular, we use the affine coordinates of adjacent corners \( O_1 = (\alpha_1, \beta_1) \), and \( O_2 = (\alpha_2, \beta_2) \) to describe the incoming and outgoing lines \((OO_1) = (\alpha - \alpha_1, \beta - \beta_1)\) and \((OO_2) = (\alpha - \alpha_2, \beta - \beta_2)\) at a corner \(O\) in affine space (see Figure 4). This allows us to compute the included angle \(\theta\) in an affine-invariant way as

\[
\cos \theta = \frac{OO_1 \cdot OO_2}{|OO_1| |OO_2|}
\]

and the bisector \((OO_3) = (\alpha - \alpha_3, \beta - \beta_3)\) as

\[
OO_3 = \frac{|OO_1| + |OO_2|}{|OO_1| + |OO_2|}
\]

(5)

to give its orientation \(\psi\) also in an affine-invariant way as

\[
\cos \psi = \frac{OO_1 \cdot OO_3}{|OO_1| |OO_3|}
\]

(6)

Thus a feature \(f\) is represented by the following affine-invariant attributes with respect to a basis \(B\) to form a 4-dimensional vector

\[
f_{B} = \langle \alpha, \beta, \theta, \psi \rangle
\]

(7)

### 4.2 Building a class model

Consider now a class of images depicting a single object under varying pose and illumination conditions. Let \(C\) be an object category. Let \(I_1, I_2, ..., I_N\) be member images depicting the objects in category \(C\). Let \(B_1, B_2, ..., B_l\) denote basis triples. Let \(f_1, f_2, ..., f_m\) denote location-specific features extracted from images. The basis frame used for describing feature \(f_i\) is given by \(B(f_i) = B_m\) for some \(m\). Let \(I(f_i)\) and \(I(B_i)\) represent the image ID of the image that contains the feature \(f_i\) and basis \(B_i\) respectively. Each feature \(f_i\) is an \(M\)-dimensional affine-invariant vector = \(< a_1, a_2, ..., a_M >\) (In Equation 7, this is a 4-dimensional vector). By quantizing each of the feature attribute values, and recording the basis triples that gave rise to them, we produce a discrete affine kernel for class \(C\) as

\[
H_{fC}(a_1, a_2, ..., a_M) = \{ B_{j_1}, ..., B_{j_k} \}, h_C >\]

(8)

where \(B_{j_1}, ..., B_{j_k}\) are the basis frames using which a feature from any of the images of the category hashed to the cell indexed by \((a_1, a_2, ..., a_M)\) and

\[
h_C(a_1, a_2, ..., a_M) = \sum_{j=1}^{\infty} \frac{n_j}{N_c}
\]

(9)

where

\[
n_j = \begin{cases} 1 & \exists B_i \in H(a_1, a_2, ..., a_M) \ \& \ I(B_i) = l_j \\ 0 & \text{otherwise} \end{cases}
\]

(10)

Thus \(n_j\) represents all the images of the class which contain a basis frame \((B_j)\) that cause a feature \(f_i\) to possess coordinates \(< a_1, ..., a_M >\) that land it in cell \(H(a_1, a_2, ..., a_M)\).

Thus \(h_C(a_1, a_2, ..., a_M) = 1\) implies that this feature is present in all of the image of the class and hence can be potentially discriminatory for the class. Further, the basis \(\{B_{j_1}, ..., B_{j_k}\}\) in
the cell form potentially corresponding basis across the images. The correspondence can be strengthened if enough such high scoring features can be found that have the same basis. In fact, by retaining the high scoring cells of the affine kernel, we get a distinct affine kernel for each class which can now serve as the class model. Thus a class model can be characterized as the set of M-dimensional affine kernel cells

$$H_C = \{ H_C'(a_1,a_2,...,a_M) | h_C'(a_1,...,a_M) > \tau \}$$  \hspace{1cm} (11)

Note that the above model of a category has similarities with both the hash table of geometric hashing [2] and the histograms in the bag of word models [10]. Unlike both geometric hashing and histograms though, the above category model explicitly captures the common occurrence of features across the images of a class with respect to a choice of basis frame. Further, since cells with

$$h_c \leq \tau$$

are discarded, this effectively eliminates distracting features from the background unless all training images of a class have similar background. Thus the model has a built-in robustness to background clutter.

By restricting the basis to lie along curves, and by computing affine coordinates of features along the same curve, we reduce the number of basis triples as well as the number of affine coordinates to be essentially linear and quadratic in the number of features. Further, since each curve is likely to come from a single object, this representation helps capture local spatial layout constraints. The resulting set of retained affine kernel cells tend to be relatively sparse.

Figure 5. Illustration of changes in affine-invariant attributes due to intra-class variations within an object category.

4.3 Modeling intra-class variations

So far we have assumed that different object instances in a category can be modeled using an affine deformation of a basic shape. From the examples in Figure 1 and 2, it is clear that the set of tolerable changes in spatial layout have to exceed those that can be modeled by rigid body deformations. Figure 5 shows an example curve which represents another object belonging to the same category as the object depicted in Figure 4 by curve C2. For ease of discussion, only the corner feature O has moved to a new location O’ which still induces a non-linear deformation that violates affine invariance. The corresponding change to the attributes is

$$f_B' =< \alpha', \beta', \theta', \psi'>$$

which is a region in the neighborhood of the original cell as shown in Figure 5. Thus a match to a feature may have to be searched in a successively wider area surrounding the original cell in the affine kernel. In general, since such changes in attribute values can be induced by a change in any features including the basis features, a systematic but progressively wider search can be achieved by dividing the affine kernel into L levels of resolution, 0..L along each feature attribute. Then the grid at level l has 2^l cells along each attribute dimension, for a total of 2^LM affine kernel cells. Here, the finest quantization is represented by l=L and the coarsest by l=0.

Thus the multi-scale affine kernel for class C at level l is given by

$$H_{f_C}^l(a_1,a_2,...,a_M) = < B_{j_1}...B_{j_k} > ^l, h_C^l >$$  \hspace{1cm} (12)

where \{ B_{j_1}...B_{j_k} \}^l are the merged affine basis that lets a feature hash into the cell at level l. The score

$$h_C^l(a_1,...,a_M)$$

is defined similarly to Equation 9 and accounts for each image occurrence only once even if multiple basis triples and their features hash to the same cell from a given image.

Since matches at lower levels of resolution can be arbitrary, they are appropriately weighted to reflect their lower contribution during matching as described next.

4.4. Class recognition using multi-scale affine kernels

We now turn to the problem of category recognition using the learned affine kernels. Given a new image depicting an object instance, we extract features using an identical process to that used in class model generation above. Thus we record the corners features and their affine-invariant attributes given in Equation 7 with respect to basis triples chosen from consecutive corners along the curve. A similar multi-scale affine kernel is formed for the query image so that

$$H_{f_Q}^l(a_1^Q,a_2^Q,...,a_M^Q) = < B_{j_1}Q...B_{j_k}Q > ^l, h_Q^l >$$  \hspace{1cm} (13)

where \{ B_{j_1}Q...B_{j_k}Q \}^l are the basis frames using which a feature

$$f_Q^l$$

from the query image hashed to the cell indexed by

$$(a_1^Q,a_2^Q,...,a_M^Q)$$

and

$$h_Q^l(a_1^Q,...,a_M^Q) = 1$$

(binary mask effectively).

To estimate how well a query image belongs to a given class C, we employ the odd-man-out principle. Specifically, we take as positive evidence when the common features present in class are also present in query. We estimate the discriminability of common features by weighing their $h_C$ score with the $h_C$ scores (frequency of occurrence) for the same feature in other classes C’ (in a manner similar to document retrieval). Also in keeping with the odd-man-out principle, we penalize the cases where the common features are absent from query.

If the query image depicted a member of the same object category as the class C, then the affine-invariant attributes of its features with respect to chosen basis will index to the same cells (this is conventional geometric hashing [2]) making the basis triples
corresponding. Let $B_q$ be a query basis triple. Consider the query features $f_{q1}, \ldots, f_{qN}$ and their affine-invariant attributes computed with respect to the basis triple $B_q$. Let $A_{B_q} = \{H_{f_{q1}}, \ldots, H_{f_{qN}}\}$ be the set of cells that are hashed by the affine-invariant attributes from these features for basis triple $B_q$. Let $H_C$ be the active cells of the multi-scale affine kernel model for the class $C$. Then $A = A_{B_q} \cap H_C$ represents the set of cells in which query features agree with common features of class $C$. Further, $B = \overline{A_{B_q} - H_C}$ represent the set of common features of class $C$ that don’t have a match in query $Q$. Finally, $qB = \overline{A_{B_q}}$ represents the set of query features that are not matched in class $C$. Since we only retained common features of class $C$, constitutes the irrelevant set based on the odd-man-out principle and hence can be ignored.

Thus the similarity between query image $Q$ and the class $C$ at level $l$ with respect to the given basis $B_q$ is given by

$$d_{B_q}(Q,C) = \frac{1}{|A|} \sum_{x \in A} h_C^l \prod_{c}(1-h_C^l) - \frac{1}{|B|} \sum_{x \in B} h_C^l \prod_{c}(1-h_C^l)$$

(14)

The term $\prod_{c}(1-h_C^l)$ represents the discriminability of the common feature in $H_C^l$.

Now we combine information across the multi-resolution matches. Since the number of images that have a feature that hash to a cell only increase with the size of the cell, the value of $h_C^l$ also includes the score from the finer level $l+1$. As is popular with pyramid kernels [14], we use an exponential weighting of $1/2^{l-1}$ for level $l$ to get the multi-level similarity between query $Q$ and class $C$ with respect to query basis $B_q$ as

$$d_{B_q}(Q,C) = \sum_{l=0}^{L} \frac{1}{2^{L-l}} d_{B_q}^l(Q,C)$$

(15)

The overall multi-level similarity between query $Q$ and class $C$ among all query basis is then given by

$$d(Q,C) = \max_{B_q} \{d_{B_q}(Q,C)\}$$

(16)

Finally, the best overall matching class for a query image $Q$ is given by

$$\hat{C} = \arg \max_{C} \{d(Q,C)\}$$

(17)

### 4.5. Category recognition algorithm

Putting it all together, the algorithm for object category recognition from scene images consists of a learning step and a classification step.

**Category Learning:**

1. Identify a set of training images depicting objects belonging to category.
2. Perform edge detection and curve extraction.
3. Extract curvature change points along the curve as corners. Add corners at junctions per line pair.
4. Identify consecutive points along the curve for basis computation.
5. Compute affine-invariant attributes (affine coordinates, angle, bisector orientation) for each corner features along the same curve.
6. Divide the range of affine-invariant attribute values per dimension to give $L$ levels of resolution.
7. Hash all affine-invariant attributes at each level of resolution and record the underlying basis and the number of images that have a feature hash to a cell.
8. Retain those cells that have a $h_C > \tau$ to form the multi-scale affine kernel for the class.

**Category Recognition:**

Given multi-scale affine kernels, one per category, and a new query image depicting an object of a category,

1. Process a query image using steps 1-6 of category learning described above.
2. For each candidate basis $B$, create a mask query affine kernel as given by Equation 13.
3. For each class $C$
   a. For each query basis $B$, count the number of cells retained by the mask for $B$ that have an overlap with the affine kernel of $C$.
   b. Retain those basis $B'$ with a cell count $> T$.
   c. For each retained basis $B'$, compute $d(Q,C)$ as per Equation 15.
   d. Compute the maximum over retained basis as per Equation 16.
4. Find the maximum match using equation 17 overall all classes to find the matching class.

In the above algorithm, Steps 3b is an efficient way to implement the maximization over the basis triples described in Equation 17.

### 4.6. Complexity Analysis

We now analyze the complexity of each of the operations in the above algorithm. Let there be $P$ classes, and each class has $N_p$ training images. Let there be $K_q$ features per training image. Since the basis triples are consecutive points along a curve, the number
The complexity of the learning stage is then \(\mathcal{O}(\sum_{i=1}^{P} \sum_{j=1}^{N} K_{ij}^2) = \mathcal{O}(P, N K^2)\). Typically, \(P < 50\), \(N < 50\), \(K < 200\) for the class of images and categories we have tested. Since the size of the retained cells in the affine kernels \(H_{i} \ll \mathcal{O}(N K^2)\), the category recognition process is fairly short, requiring computations of \(O(K^2)\) for query attribute generation for \(K\) query basis, and \(O(K) \mathcal{O}(H_{i})\) per class for a total of \(\mathcal{O}(\sum_{i=1}^{P} K | H_{i}|)\).

In practice, the category learning step takes a few minutes for a small class of images (<50), and the category recognition takes milliseconds on a Pentium 4, 2 GHz, 1g RAM machine.

4.7 Discussion

The above category recognition method exploits the ideas of geometric hashing and combines them with class category recognition. Like geometric hashing, we extract affine-invariant features, although our features are 4-dimensional. Unlike geometric hashing which takes a histogram of basis triples to find candidate matches for a query triple using the information recorded in the hash table, our method only uses the hash table for class membership. Our method can also be contrasted with other object category recognition methods. For example, similar to the methods in document retrieval that use term-document-inverse-document frequencies of words, we also model the discriminatory power of common features by noting the comparative scores for a feature across other classes (Equation 14). Our method also uses multi-resolution kernels similar to pyramid kernels[14]. However, the multi-resolution is more for handling within-class object appearance variations rather than pose changes such as scale or rotation (which are automatically handled using affine-invariant coordinates). Finally, although we have demonstrated the method with corner features, any affine-invariant features including those given by SIFT detector [18] could be substituted without materially affecting the principle of categorization.

<table>
<thead>
<tr>
<th>Cell phone class images</th>
<th>Number of Points</th>
<th>Number of Basis</th>
<th>Number of Curves</th>
<th>Number of affine GQs</th>
<th>Number of retained cells with (h_{i} \neq 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>177</td>
<td>160</td>
<td>45</td>
<td>265</td>
<td>13</td>
</tr>
<tr>
<td>2.</td>
<td>260</td>
<td>249</td>
<td>76</td>
<td>402</td>
<td>13</td>
</tr>
<tr>
<td>3.</td>
<td>88</td>
<td>86</td>
<td>38</td>
<td>176</td>
<td>13</td>
</tr>
<tr>
<td>4.</td>
<td>241</td>
<td>229</td>
<td>84</td>
<td>380</td>
<td>13</td>
</tr>
</tbody>
</table>

Table 1. Illustration of feature reduction in the multi-scale affine kernel.

5. RESULTS

We now present results of experiments of object categorization on large image data set. We begin by illustrating our method with an example. Figure 6a shows sample images of the cell phone object category. Here the object can be seen against different backgrounds and under different poses. Each instance of the cell phone in the images shown comes from a different cell phone maker. Hence there is no single rigid-body transformation that can align one cell phone to the other. Nevertheless, they possess common features in a majority of locations, eg. a rectangular dial, dial keypad keys, etc. The set of curves and corner features extracted from each image are shown in Figure 6b (corners are
indicated by dots). As can be seen, the features capture meaningful structural information on objects and the background. Since the basis and features are both computed along curve groups, the number of affine-invariant feature attributes computed is roughly linear in the number of features. This can also be observed from Table 1 where the number of points and affine coordinates (affine-invariant attributes) are roughly the same. In this table, the images referred as 1 through 4 are the images of Figure 6a left to right, top to bottom. The size of the affine kernel without intra-class feature accounting could be large. For example, for the 4 image class of Figure 6, there are a total of $265+402+176+380=1223$ possible cells. Of these, only 13 of the cells have a $h_c = 1$. Thus if we retain only the features points that occur in all 4 images, we would retain only 13 of the 1223 cells of the affine kernel for this class. In general, we use a minimum threshold $\tau = 0.6$ retaining cells with features common to 60% of the images in the training category. Further, we use ten levels of resolution for quantization following the popular choice in multi-resolution representations [14]. The set of 13 most common features across the 4 images of the cell phone class of Figure 6a are shown in Figures 6c overlaid in red. It is interesting to note that all of these features came from the foreground object. Thus the background clutter is effectively eliminated in the multi-scale affine kernel representation.

5.1 Classification results
We tested the performance of our approach on the Caltech-101 dataset[15]. This is a benchmark dataset of 101 diverse object categories and is widely used for evaluating class recognition methods. It was collected by manually filtering the results of keyword-based Google image search. The number of images vary from 31 to 800 (for airplanes) images per category for a total of over 8000 images. This dataset is good for object recognition as the objects often occupy a large part of the image although the backgrounds still vary. We tested our method for each of the Caltech-101 dataset categories except the background Google category which shows an assortment of backgrounds. Since we use a threshold of $\tau = 0.6$ for common features, this resulted in a number of images being dropped from the categories. For example, the cell phone category resulted in 3 sub-categories, namely, one-piece open, flip-top and closed phones (front-view). Since the dominant number of images were for the open one-piece phone (38 versus 5 and 7), we retained only this subcategory. When there were sufficient training images for a subcategory, we retained them as well, so that a category could be represented by more than one class of images. The accuracy numbers in that case were added per class.

Figure 7 illustrates the classification results for a set of categories. The query image is shown on the left. The matching category is shown using 3 representative member images. The corresponding feature map is shown below the images. As can be seen, the best matching category has been identified even though the individual object instance shows considerable variation in appearance, pose and background. Table 4 second row, list the classification accuracy for a representative set of classes.

5.2 Comparison with other approaches
To compare the performance of our method with the state-of-the-art, we implemented two other popular approaches for object category recognition. Specifically we implemented the bag of features model described in Sivic et al. [11] and the pyramid kernels[14]. To measure the accuracy of classification, we used 98 (dropped the miscellaneous and faces categories for obvious reasons) object classes from Caltech data set with both rigid (e.g. airplane), articulated (e.g. scissors) and nonrigid objects (e.g. butterflies). For each class, we use 20 training images for training and the remaining for testing. The set of images used for training and testing was made the same for all three methods. As required for the bag-of-features[11], we used SIFT features for describing images. A vector quantization scheme was then used to form a codebook of features per image. These were fed to the probabilistic latent semantic analysis (pLSA) model for recognition. Next, we obtained an implementation of pyramid kernels [14]. Since the pyramid kernel approach works with any choice of multi-dimensional features, we used the same features as in our approach. Specifically, 10-level pyramid kernels were constructed from the raw corner features (not affine-invariant attributes) and were used to train an SVM classifier.
category against others using bag of features. This matrix shows that the between class distinction achieved by the bag of words model is low in most cases varying from 6% to 69% on the average as shown in the last row of this table. In comparison, the confusion matrix using our method is shown in Table 3. As can be seen, the recognition performance is better using our approach for all the classes. Further, the confusion between classes is reasonable as expected. For example, the umbrella class and the chandelier are confused as they can be regarded as upside down umbrellas. The scissors class had the lowest recognition rate using our method as it possesses the least number of features in comparison to the background. In Table 2 and 3, the best matching class is highlighted in red.

The recognition rates achieved using the pyramid kernels for the same set of target classes are shown in Table 4 row 1. The average recognition was 63% across all the classes using the pyramid kernels. The recognition rates are averaged over 100 runs, where training and testing samples are randomized over the runs. In comparison, the comparison accuracy using our method is shown in row 2 of Table 4. In most categories, we found our method improved upon pyramid kernels as it takes into account pose changes and intra-class object variations better. The class of airplanes is one of the anomalies we have found where the background features dominate the foreground. We also observed no measurable difference in performance across rigid and non-rigid object categories (cell phone, ceiling fan, scissors) vs Lotus flowers.

### 6. CONCLUSIONS

In this paper we have presented a novel algorithm for object category recognition that closely approximates visual object grouping using the odd-man-out principle. The resulting classification is not only of higher accuracy but also gives plausible explanation for why a particular image depicting an object belongs to a category even with the choice of simple features. Future work will explore richer features for image representation including color and texture.

---

### Table 2. Illustration of confusion matrix for bag-of-features method.

<table>
<thead>
<tr>
<th>Category</th>
<th>Airplane</th>
<th>Anchor</th>
<th>Butterfly</th>
<th>Ceiling Fan</th>
<th>Cellphone</th>
<th>Chandelier</th>
<th>Lotus</th>
<th>Pyramid</th>
<th>Scissors</th>
<th>Umbrella</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>0.50</td>
<td>0.1</td>
<td>0.3</td>
<td>0.15</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.1</td>
<td>0.2</td>
<td>0.15</td>
</tr>
<tr>
<td>Anchor</td>
<td>0.1</td>
<td>0.67</td>
<td>0.06</td>
<td>0.12</td>
<td>0.07</td>
<td>0.05</td>
<td>0.01</td>
<td>0.1</td>
<td>0.2</td>
<td>0.01</td>
</tr>
<tr>
<td>Butterfly</td>
<td>0.01</td>
<td>0.07</td>
<td>0.916</td>
<td>0.08</td>
<td>0.23</td>
<td>0.22</td>
<td>0.12</td>
<td>0.01</td>
<td>0.01</td>
<td>0.12</td>
</tr>
<tr>
<td>Ceiling Fan</td>
<td>0.12</td>
<td>0.11</td>
<td>0.01</td>
<td>0.923</td>
<td>0.03</td>
<td>0.25</td>
<td>0.01</td>
<td>0.12</td>
<td>0.11</td>
<td>0.12</td>
</tr>
<tr>
<td>Cellphone</td>
<td>0.032</td>
<td>0.014</td>
<td>0.01</td>
<td>0.0043</td>
<td>0.645</td>
<td>0.12</td>
<td>0.03</td>
<td>0.15</td>
<td>0.067</td>
<td>0.18</td>
</tr>
<tr>
<td>Chandelier</td>
<td>0.023</td>
<td>0.056</td>
<td>0.34</td>
<td>0.18</td>
<td>0.21</td>
<td>0.78</td>
<td>0.052</td>
<td>0.012</td>
<td>0.01</td>
<td>0.34</td>
</tr>
<tr>
<td>Lotus</td>
<td>0.02</td>
<td>0.09</td>
<td>0.21</td>
<td>0.09</td>
<td>0.043</td>
<td>0.033</td>
<td>0.008</td>
<td>0.016</td>
<td>0.08</td>
<td>0.32</td>
</tr>
<tr>
<td>Pyramid</td>
<td>0.19</td>
<td>0.21</td>
<td>0.09</td>
<td>0.11</td>
<td>0.14</td>
<td>0.051</td>
<td>0.12</td>
<td>0.045</td>
<td>0.32</td>
<td>0.15</td>
</tr>
<tr>
<td>Scissors</td>
<td>0.12</td>
<td>0.92</td>
<td>0.32</td>
<td>0.067</td>
<td>0.19</td>
<td>0.24</td>
<td>0.32</td>
<td>0.69</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>Umbrella</td>
<td>0.13</td>
<td>0.94</td>
<td>0.13</td>
<td>0.21</td>
<td>0.09</td>
<td>0.23</td>
<td>0.124</td>
<td>0.17</td>
<td>0.21</td>
<td>0.745</td>
</tr>
<tr>
<td>Avg.</td>
<td>0.78</td>
<td>0.52</td>
<td>0.9</td>
<td>0.92</td>
<td>0.84</td>
<td>0.91</td>
<td>0.64</td>
<td>0.69</td>
<td>0.75</td>
<td></td>
</tr>
</tbody>
</table>

---

### Table 3. Illustration of confusion matrix using our method.

### Table 4 Comparison of recognition rates with the Pyramid match kernel method.

<table>
<thead>
<tr>
<th>Class</th>
<th>Airplane</th>
<th>Anchor</th>
<th>Butterfly</th>
<th>Ceiling Fan</th>
<th>Cellphone</th>
<th>Chandelier</th>
<th>Lotus</th>
<th>Pyramid</th>
<th>Scissors</th>
<th>Umbrella</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pyramid</td>
<td>100%</td>
<td>45%</td>
<td>49.8%</td>
<td>78.1%</td>
<td>91.5%</td>
<td>69.1%</td>
<td>26.8%</td>
<td>62.5%</td>
<td>94.0%</td>
<td>53.3%</td>
</tr>
<tr>
<td>Affine</td>
<td>50%</td>
<td>57%</td>
<td>91.8%</td>
<td>92.3%</td>
<td>94.5%</td>
<td>78.7%</td>
<td>90.8%</td>
<td>84.5%</td>
<td>69%</td>
<td>74.5%</td>
</tr>
</tbody>
</table>
REFERENCES


