Visual Object Detection using Frequent Pattern Mining

Yousuf A and **Ravindran B**
Department of Computer Science and Engineering,
Indian Institute of Technology Madras
yousufa@cse.iitm.ac.in, bravi@cse.iitm.ac.in

Abstract

Object search in a visual scene is a highly challenging and computationally intensive task. Most of the current object detection techniques extract features from images for classification. From the results of these techniques it can be observed that the feature extraction approach works well for single images but are not sufficient for generalizing over a variety of object instances of the same class. In this work we try to address this problem by using a well known machine learning technique, namely, *frequent pattern mining.* The approach we use here is to find frequently occurring patterns of visual properties across the whole set of images in the class. The frequent patterns thus found would potentially represent those features which bind together the images of that class. Shape, color, texture, spatial orientation etc., or any combinations of these can be used as the visual properties. During the testing phase the object presence is detected by analyzing the images for the presence of these learned patterns. The proposed framework has been tested with Caltech 101-object dataset and the results are presented.

Introduction

Visual object detection is the task of identifying an object from a visual scene, for example, identifying a pen on a table or a book in a book shelf. Humans do a lot of object detection in daily activities like reading a book, crossing the road, identifying persons etc. It can be seen that humans use the visual input heavily to interact with the environment. Thus object detection is an important task to be solved for the goal of building autonomous agents that visually interact with their surroundings.

Consider the task of describing an object, say, a cup. The description should possibly include all different cups that we know (for example as shown in figure 1). Soon we realize that it is very hard to come up with a description that includes all varieties of cups. Thus to have a description that generalizes over all the object instances within a class is very difficult and this makes object detection a hard problem. Image processing is also equally hard because for reliable detection we have to handle issues like scale invariance, different lighting conditions, object occlusion, etc. Due to these reasons it is hard to make a rule based system and the most ideal approach would be to learn the solution mechanism tailored to the task.

Figure 1: Different varieties of cups

An autonomous agent may encounter new object classes that should be learned on the go. It is highly unlikely that all object classes (labels) would be present in the initial training phase of the agent and the system should be able to learn new labels as needed. This requires having an incremental vision system to which new labels can be added with minimal relearning effort. This is again a significant challenge to the learning system as many machine learning techniques are naturally not incremental.

In this work we describe an object detection system which tries to extract the common visual properties among a set of images. The approach is motivated by Shimon Ullman's visual routines theory (Ullman 1984) which states that the human vision system is composed of basic visual operators which are combined in different ways for complex object detection tasks. Visual operators are operators which can extract specific visual properties like color, shape, texture, etc. Thus the learning phase for an object class tries to find out the common visual properties that occur in the set of images (within in the class) and these set of visual properties are later used to detect an object in the testing phase.

The rest of this paper is organized as follows. The next section talks about related work. This is followed by our approach, formal definition and algorithms. Finally we conclude by presenting experimental results and future work.

Related Work

Research on object detection can be broadly classified into four groups which are as follows. One line of research is to consider the task as a supervised classification problem where the agents tries to classify the image pixels as foreground (fg) and background (bg) pixels based on features extracted from the image. The state of the art technique that uses this approach is SIFT (Lowe 2004) or scale invariant feature transform. SIFT learns features which are invariant to image scaling, translation, and rotation. It creates what the author calls *image keys* which allow local geometric deformations and these are used as input to a nearest neighbor indexing that identifies candidate object matches.

Another line of approach in object detection is to consider part-based representation for objects. (Felzenszwalb, McAllester, and Ramanan 2008) describes object detection system which uses multiscale deformable part models. In this model, each part encodes local appearance properties of an object and the deformable configuration is characterized by spring like connections between certain pair of parts. (Agarwal and Awan 2004) describes object detection using a sparse part-based representation. Here a vocabulary of distinctive object parts are automatically constructed from the set of sample images of the object class and the objects are represented using this vocabulary together with the spatial relation among these parts.

Researchers have also tried developing a visual grammar for object representation. The visual grammar describes an object using variable hierarchical structures. The object will be represented in terms of parts and each part is again described by sub-parts or by properties of that part. (Zhu and Mumford 2006) explains such an approach but the major difficulty is defining what constitutes the various parts.

Many hybrid systems have also been proposed. (Frintrop 2006) describes a visual detection system in which the object training and detection happens in two phases. Initially the object class is trained to find out how the object "stands out" from its surroundings. This is done by learning the weights for various Gabor filters which responds to variations in intensity, color and orientation. These potential object areas are later fed to a Viola-Jones classifier which identifies the features of the object.

There has been work from the machine learning community with emphasis on learning visual operators for object detection and gaze control. Minut and Mahadevan (Minut and Mahadevan 2001) have proposed a selective visual attention model based on reinforcement learning in which the agent has to choose the next fixation point (action) based only upon the visual information.

Mohan et. al. (Sridharan, Wyatt, and Dearden 2008) has developed a hierarchical POMDP framework called HiPPO which uses various shape and color visual operators for addressing object identification queries. The model addresses a variety of visual queries like location of the object, number of objects in the scene, identifying the type of the object, etc. For each query HiPPO does an offline planning on the region of interest (ROI) and finds a set of operators required to answer the query. The result of their experiments have shown that it is computationally more efficient to plan a set of operators for the visual query rather than doing the naive exhaustive search. In order to add new labels to HiPPO, it would require reformulating the model with new states, actions, and observations and do re-planning on the query.

In this work inspired by HiPPO and visual routines theory, we propose an object detection system that uses frequent pattern (FP) mining. Frequent pattern mining is a well known technique in data mining. It was initially used for market basket analysis to find out the frequently occurring items in purchase transactions. The set of items present in a transaction is termed as the *itemset* and the set of itemsets forms the *transaction database.* The transaction database is analyzed for the occurrence of patterns in the items and the co-occurrence of these patterns. This helps to better understand customer purchase patterns. One of the major contributions of this work is the application of frequent pattern mining for object detection. So far there has been little work done on pattern mining in images for object detection. The details of the system are explained in the next section.

Our Approach

We assume that we have a set of labeled images with background and foreground information for training. We define a set of visual operators O which includes property operators P and relational operators R. The property operators detect various visual properties from the input image like shape, color, etc. It is defined as an operator which checks for a particular property in an image and returns all instances of that property as the set of pixels from which the properties were observed. For example, a circle operator applied on an image should return all circles in the image with the location of each circle. Relational operators define the various relations among the instances identified by the property operators. Spatial operators like *left, right, inside* and *outside*, etc. are examples of relational operators. The working of the proposed system can be logically partitioned into three phases viz. the frequent pattern mining phase, selection of operators and re-scoring of operators.

Frequent Pattern Mining phase

Analogous to the FP mining terminology, we define a visual operator o as an item. A set of visual operators O forms an itemset. The visual operators identified for an image forms a visual transaction and the set of visual transactions identified for all the images in a class forms the visual database of that particular class (see figure 2).

The FP mining phase starts by constructing the visual transaction database for each class and finds the frequent patterns for each class. A property operator is said to have *fired* on an image i.e., the operator is included as an item in the visual transaction of an image, if it selects atleast some minimum count of pixels from the foreground. All available property operators are applied on an image and those which fire are found. The relationship among the instances identified by the property operators are determined by applying the relational operators on them. Thus the property operators identified, together with the relational operators forms the visual transaction for that image. This procedure is re-

Figure 2: Shows the Property operators, Relational operators and the visual transaction for stop sign image

peated for all available images in the class and the identified visual transactions constitute the visual database of the class.

Apriori frequent pattern mining (Agrawal, Imielinski, and Swami 1993) is used on these transactions to find the frequently occurring visual itemsets. The candidate generation is done on the apriori property that, for an itemset to be a frequent itemset candidate, all its subsets should also be frequent. Similarly the frequent patterns are found for each class independently and these FPs forms the input to the scoring phase. Apart from fixing the values of parameters like minimum-support, thresholds for visual operators, etc., no user intervention is required in this phase. Algorithm 1 outlines the steps required for a single class. Routine *find_relational_opers* is used to find the relational operators among the identified property operators and pseudo-code is not shown as it is similar to the code for property operators.

Algorithm 1: FP Mining

Input: label L, Training Data T
Output: set of frequent operators, FP

for *each image i and foreground info.* $f \in T$ **do**
    $O[i] \leftarrow NIL$;
    for *each property operator* $p \in P$ **do**
        fgpixels ← apply_operator(p, i, f);
        if *fgpixels* > *select_threshold* **then**
            $O[i] \leftarrow O[i] \cup p$
    O[i] ← O[i] ∪ find_relational_opers(O[i], R);
FP ← Apriori_FPMine(O, minsupport);

Scoring of frequent operators

While adding an operator in the frequent pattern mining phase, we considered only the coverage of the operator. The FP mining procedure ensures that the patterns obtained would be present atleast in *minimum-support count* images of the class. But this does not ensure how accurate the operator sets are. The operator may select more background pixels than foreground pixels or the coverage of this operator might be superseded by some other operator (see figure 3).

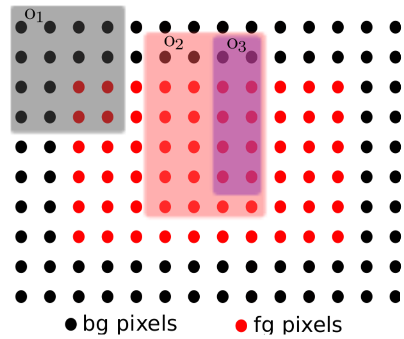


Figure 3: o_1 has relatively lower score, o_2 supersedes o_3's coverage

For pruning such inefficient or extra operators from the set of identified frequent operators, we attach a score with each of the property operators. The score is defined as the accuracy measure, i.e. the ratio of true positives (tp) and true negatives (tn) detected by an operator to the total pixels of the image.

$$\text{score} = \frac{\text{tp} + \text{tn}}{\text{total pixels}}$$

The true positives for an operator are those pixels selected by the operator which are true fg pixels (as per the object foreground information) and the true negatives are the true bg pixels. From the identified set of frequent operators (FP), we have to find those subsets which give the maximum score (FP_{max}) for the class. This is done by constructing a lattice of frequent operator sets by considering inclusion on the FPs (see figures 4, 5).

Each node in this lattice is frequent because they themselves are subsets of the frequent itemsets and is associated with a score. The score is computed as the average accuracy of the operator set computed across all the images of that class. The search for the maximum-score operator set starts from the first level (single operators). In each level the operator set which has the maximum score (maximal operator set) is found. This maximal operator set is compared with the maximal operator set of second level and so on. The search continues till no improvement in the maximum score is seen i.e., till the score of i^{th}level is less than or equal to $i\text{-}1^{th}$level. Once the maximal operator set is found out, it is included to the FP_{max} set and the maximal operator set along with all of its subsets and supersets are not further considered in the search process. The maximal operator set covers for its supersets and its subsets and hence they are not further analyzed. The above steps are repeated until all the nodes of lattice are removed. Algorithm 2 enumerates the steps involved in scoring the frequent patterns. Routine *find_max* finds the maximum scored operator set among all the operator sets in the specified level.

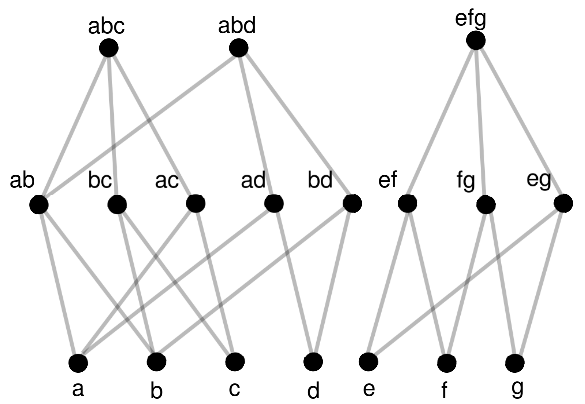


Figure 4: Lattice constructed for 3 operator sets *abc*, *abd* and *efg*

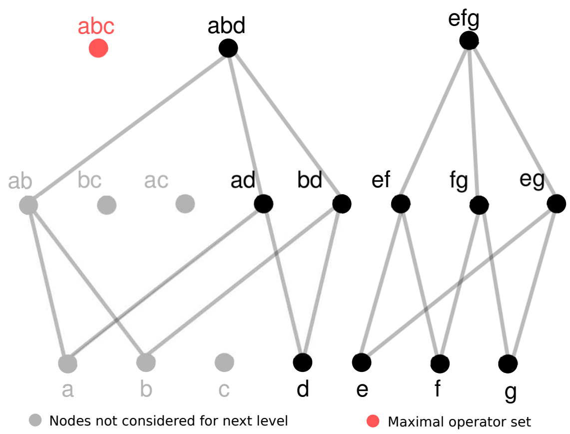


Figure 5: Lattice considered for second iteration of algorithm, maximum operator set in first iteration is *abc*

Rescoring of operators

So far we have processed object classes independent of each other and have not considered the incremental requirement of the system. While adding new classes to the system, it might happen that the operator set with the maximum score identified for one class would be popular for other classes too. For example a circle operator will have a good score for detecting a pizza class and a soccer ball class.

In such cases the "popular operators" must be switched with other alternatives (if present). This has to be done in an online fashion as all the object classes may not be present during training. Thus in the pizza-soccer ball example, the system should identify a slightly less scored frequent operator set which detects the various hexagons and pentagons in the soccer ball. This rescoring of the operator sets should be done as and when new labels are added to the system.

From the previous scoring phase, we have identified the frequently occurring maximum scored operator sets. From the frequency of occurrence of an operator set in a class, we can calculate the probability of the operator set given the label i.e $P(\mathrm{FP}_i|\mathrm{L})$ where L is a label and FP_i a frequent

Algorithm 2: Scoring the FPs

```
Input: set of frequent operators, FP
Output: set of operators, FP_max
construct the lattice for FP ;
while there are still nodes in the lattice do
    max ← find_max(O_1, T) ;
    for level ← 2 to max do
        max_curr ← find_max(O_level, T) ;
        if max_curr.score ≤ max.score then
            break ;
        max ← max_curr ;
    FP_max ← FP_max ∪ max.op ;
    Remove max, its subsets and supersets from
    further consideration;
```

operator set. This probability can be written as,

$$P(\mathrm{FP}_i|\mathrm{L}) = \frac{N_{\mathrm{FP}_i}}{\Sigma_i N_{\mathrm{FP}_i}}$$

where N_{FP_i} is the number of images FP_i has fired for the class.

For testing the presence of a label in an image, we require the probability of a label given that we observed a particular operator set. Using Bayes rule, we can find the probability of label given the observation of a frequent pattern FP_i as,

$$P(L|\mathrm{FP}_i) = \frac{P(\mathrm{FP}_i|\mathrm{L}) * P(\mathrm{L})}{P(\mathrm{FP}_i)}$$

P(L) is the probability of the label which is assumed constant, given by N^L/T where N^L is the number of images of the class L, and T is the total number of images across all classes. We assume equal number of training images for all classes i.e, $N^{L_i} = N^{L_j}$. With the above assumption $P(\mathrm{FP}_i|\mathrm{L})$ can be rewritten as $N^L_{\mathrm{FP}_i}/N^L$. The probability of observing a frequent pattern $P(\mathrm{FP}_i)$ is N_{FP_i}/T i.e., the number of images on which FP_i fired regardless of the label, divided by the total number of images. Substituting all of these in the above equation we have,

$$\begin{aligned}
P(\mathrm{L}|\mathrm{FP}_i) &= \frac{P(\mathrm{FP}_i|\mathrm{L}) * P(\mathrm{L})}{P(\mathrm{FP}_i)} \\
&= \frac{N^L_{\mathrm{FP}_i}}{N^L} \times \frac{N^L}{T} \times \frac{T}{N_{\mathrm{FP}_i}} \\
&= \frac{N^L_{\mathrm{FP}_i}}{\cancel{N^L}} \times \frac{\cancel{N^L}}{\cancel{T}} \times \frac{\cancel{T}}{T_{\mathrm{FP}_i}} \\
\therefore P(\mathrm{L}|\mathrm{FP}_i) &= \frac{N^L_{\mathrm{FP}_i}}{N_{\mathrm{FP}_i}}
\end{aligned}$$

The above result shows that for testing a label, the operator sets should be ordered according to the ascending order of $\frac{N^L_{\mathrm{FP}_i}}{N_{\mathrm{FP}_i}}$. This for an operator set to get a better score in this phase, either the frequency of observing the operator set FP_i for the particular label is high or that the probability of the operator set FP_i firing for other classes is less. Algorithm 3 enumerates the steps.

Testing

During testing we were specifically interested in addressing queries like *"Is there an object of particular type in the image?"*. For answering this query, the test image needs to be checked for the presence of only those visual patterns that are learned for the queried label. Each operator set of the queried class is applied to the test image in decreasing order of probability value $P(\mathrm{L}|\mathrm{FP})$ until a match is obtained or all operators sets have been checked. If a match is found, we determine all labels (associated labels) for which the matched operator set fired during training. The object is considered present only if the queried label is same as the label which has maximum $P(\mathrm{L}|\mathrm{FP})$ value from the set of associated labels. For computing the associated labels, each operator set is mapped with a dependency list which contains information of all the training labels for which the operator set fired.

Algorithm 3: Rescoring the FPs

```
Input: FP_max for all classes
Output:
for each class, c ∈ C do
    for each FP ∈ c.FP_max do
        freq[FP] ← freq[FP] + FP.freq
for each class, c ∈ C do
    sort(c.FP_max) by c.FP_max.freq / freq[FP_max]
```

Experiments

For experimental validation of our approach we have selected basic operators like *shape* (square, triangle, circle, etc.), *color* (red, green, blue) as property operators and binary spatial operators like *left, right, top, bottom, inside, distinct* etc. as relational operators (see table 1). The shape operator detects and returns all the instance of the respective shape. The color operator is used in conjunction with the shape operator, i.e., the color operator is applied to only those regions where shape operator has fired. Once we have all the shapes identified, the spatial operators are applied by considering the shapes two at a time.

The image data set we used was Caltech 101-object categories (http://www.vision.caltech.edu/Image_Datasets/Caltech101/ 2004). The results discussed here are not state-of-the art in object detection, as it can be observed that the selected operators may not work well for real world scenes. Although we have tested only with the basic operator sets, choosing the operators to detect complex features like SIFT or its variants would give better results. Out of the 101 object labels, we empirically selected a set of labels in which the operators performed well.

Figures 6, 7 and 8 shows some images from the data set, their corresponding operator fg selection marked as red and the optimal operator set learned.

The system was trained on 30 images from each class and testing was done on 15 images different from the training set. Table 2 shows the selected labels and percentage of detection accuracy, top-2, top-3 and rejection accuracy for the system build with shape operators (no spatial relation operators). The detection accuracy for a label is the percentage of true positives(TP) detected by the classifier among all the positive images (*recall*). The system was trained on all the labels, and the testing was done on all positive images, 15 images per class. The match for a particular label is given as explained in the Testing section. In top-2 and top-3 matches, the object is considered to be detected if the queried label occurs atleast in the top-2 or top-3 of $P(\mathrm{L}|\mathrm{FP})$ ordering of the fired operator set. The low scores towards the end of the table show those labels where the operator sets learned are similar (or same). We observed that in these cases there were no alternate operators to switch in the re-scoring phase.

Class	Operator	Description
Shape	triangleop, squareop, circleop, 5polyop, 6polyop,7polyop 8polyop, 9polyop	identifies shapes
Color	redop greenop blueop	identifies color
Spatial	left,right,top,bottom inside,touch,overlap,distinct	defines relation

Table 1: Visual operators

The rejection accuracy is the percentage of true negatives (TN) detected by the classifier among all the negative samples (*specificity*). The system was trained on all the labels, and the testing was done on 255 images per class, i.e., 15 images from each class except the true class. Note that this measure cannot be directly correlated with the TP%.

Similarly in table 3 the results are given for the system built with shape and spatial operators. Here we can observe that the detection rate decreases (and increases for some labels) and the rejection rate (TN rate) improves as the operators now impose the spatial constraints also for a successful match.

Conclusion

In this work we proposed a dynamically extendable visual object detection system using frequent pattern mining. To the best of our knowledge this is the first such work that learns frequent patterns from the images of a class. The results show that even with simple operators the model is able to detect a variety of fairly complex objects.

Future work

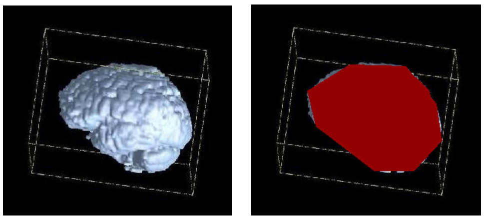
The visual operators used in the experimentation were not state-of-the-art ones and more accurate operators can be used for better detection. We are currently working on adding more complex operators like SIFT to the system. We are also working towards adding potential ROI detection thereby doing selective visual attention. Work is also been done to see if the sequence of operators that needs to be applied for detecting the presence of an object can be learned using U-Trees (Mccallum 1996).

Class	TP%	Top-2 TP%	Top-3 TP%	TN%
dollar_bill	100	100	100	56.078
pizza	73.333	73.333	73.333	66.274
stop_sign	53.333	53.333	100	99.607
lamp	53.333	80	80	75.686
ceiling_fan	53.333	53.333	60	67.058
soccer_ball	46.666	46.666	46.666	94.509
metronome	46.666	46.666	46.666	90.98
watch	46.666	46.666	46.666	81.568
sunflower	46.666	53.333	53.333	68.627
yin_yang	33.333	86.666	86.666	81.568
airplanes	33.333	33.333	93.333	81.176
strawberry	26.666	26.666	33.333	74.117
barrel	26.666	26.666	26.666	66.274
camera	6.666	6.666	6.666	97.647
brain	0	0	0	100
umbrella	0	40	40	100
accordion	0	13.333	66.666	100
scissors	0	73.333	73.333	87.843

Table 2: Selected labels from 101dataset and their detection/rejection accuracy with shape operators

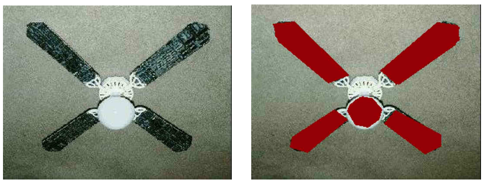
Class	TP%	Top-2 TP%	Top-3 TP%	TN%
dollar_bill	100	100	100	63.137
pizza	46.666	46.666	66.666	97.647
stop_sign	40	40	40	99.607
lamp	66.666	66.666	66.666	61.568
ceiling_fan	60	60	60	58.823
soccer_ball	46.666	46.666	46.666	99.607
metronome	66.666	66.666	66.666	74.117
watch	53.333	53.333	53.333	73.725
sunflower	46.666	53.333	53.333	80
yin_yang	86.666	86.666	86.666	46.666
airplanes	33.333	93.333	93.333	81.176
strawberry	26.666	26.666	46.666	74.117
barrel	20	26.666	26.666	81.176
camera	6.666	6.666	6.666	98.823
brain	0	0	0	100
umbrella	0	40	80	100
accordion	0	53.333	53.333	98.431
scissors	0	73.333	73.333	87.843

Table 3: Selected labels from 101dataset and their detection/rejection accuracy with shape and spatial operators



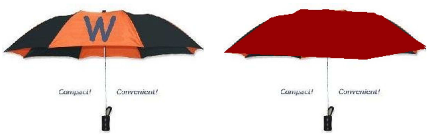
(a) brain image (b) identified mask

Figure 6: optimal operator set = 8polyop



(a) ceiling fan image (b) identified mask

Figure 7: optimal operator set = 5polyop, 8polyop



(a) umbrella image (b) identified mask

Figure 8: optimal operator set = 5polyop, 6polyop, redop

References

Agarwal, S., and Awan, A. 2004. Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. Pattern Anal. Mach. Intell.* 26(11):1475–1490. Member-Roth, Dan.

Agrawal, R.; Imielinski, T.; and Swami, A. N. 1993. Mining association rules between sets of items in large databases. In Buneman, P., and Jajodia, S., eds., *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, 207–216.

Felzenszwalb, P. F.; McAllester, D. A.; and Ramanan, D. 2008. A discriminatively trained, multiscale, deformable part model. In *CVPR*.

Frintrop, S. 2006. *VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search*, volume 3899 of *Lecture Notes in Computer Science*. Springer.

http://www.vision.caltech.edu/Image_Datasets/Caltech101/. 2004. Caltech 101 object category dataset.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60:91–110.

Mccallum, A. K. 1996. *Reinforcement learning with selective perception and hidden state*. Ph.D. Dissertation. Supervisor-Ballard, Dana.

Minut, S., and Mahadevan, S. 2001. A reinforcement learning model of selective visual attention. In *In Proceedings of the Fifth International Conference on Autonomous Agents*, 457–464. Press.

Sridharan, M.; Wyatt, J.; and Dearden, R. 2008. Hippo: Hierarchial pomdps for planning information processing and sensing actions on a robot. *International Conference on Automated Planning and Scheduling (ICAPS 2008)*.

Ullman, S. 1984. Visual routines. *Cognition* 18:97–156.

Zhu, S. C., and Mumford, D. 2006. A stochastic grammar of images. *Found. Trends. Comput. Graph. Vis.* 2(4):259–362.