7-2016

Active Object Localization in Visual Situations

Max H. Quinn
Portland State University

Anthony Rhodes
Portland State University

Melanie Mitchell
Portland State University, mm@pdx.edu

Active Object Localization in Visual Situations

Max H. Quinn, Anthony D. Rhodes, and Melanie Mitchell

***Abstract*—We describe a method for performing active localization of objects in instances of visual situations. A visual situation is an abstract concept—e.g., "a boxing match", "a birthday party", "walking the dog", "waiting for a bus"—whose image instantiations are linked more by their common spatial and semantic structure than by low-level visual similarity. Our system combines given and learned knowledge of the structure of a particular situation, and adapts that knowledge to a new situation instance as it actively searches for objects. More specifically, the system learns a set of probability distributions describing spatial and other relationships among relevant objects. The system uses those distributions to iteratively sample object proposals on a test image, but also continually uses information from those object proposals to adaptively modify the distributions based on what the system has detected. We test our approach's ability to efficiently localize objects, using a situation-specific image dataset created by our group. We compare the results with several baselines and variations on our method, and demonstrate the strong benefit of using situation knowledge and active context-driven localization. Finally, we contrast our method with several other approaches that use context as well as active search for object localization in images.**

***Index Terms*—Object Detection, Active Object Localization, Visual Situation Recognition**

I. Introduction

CONSIDER the images in Fig. 1. Humans with knowledge of the concept "Walking the Dog" can easily recognize these images as instances of that visual situation. What objects and relationships constitute this general situation? A simplified description of the prototypical "Dog-Walking" situation might consist of a human dog-walker holding a leash that is attached to a dog, both of them walking (Fig. 2). This concept prototype can be mapped straightforwardly onto the instances in Fig. 1.

In general, in order to recognize and understand a particular abstract visual situation across varied instances, a perceiver must have prior knowledge of the situation's expected visual and semantic structure, and must be able to *flexibly* adapt that knowledge to a given input. Fig. 3 shows several images semantically similar to those in Fig. 1, but which depart in some way from the prototype of Fig. 2. In some instances there are not one, but multiple dogs or dog-walkers; in some the dog-"walkers" and dogs are not walking, but running (or biking, or swimming, or otherwise riding wheeled vehicles); in some, the leash is missing. In the last, the dog-walker is not a person, but rather another dog.

In spite of these variations, most people would consider all these (sometimes humorous) images to be instances of the same abstract category, roughly labeled *Dog-Walking*. This is because people have the ability to quickly focus on relevant objects, actions, groupings, and relationships in the image, ignore irrelevant "clutter," and allow some concepts from the prototype to be missing, or replaced as needed by related concepts. For example, "leash" can be missing; "dog" can be replaced by "dog-group" ; "walking" can change to to "running," "swimming," or "riding"; "dog-walker (human)" can become "dog-walker (dog)," and so on. In [2], Hofstadter et al. defined "conceptual slippage" as the act of modifying concepts so as to flexibly map a perceived situation to a known prototypical situation. Making appropriate conceptual slippages is at the core of analogy-making, which is central to the recognition of abstract concepts. [3]

In general, a *visual situation* defines a space of visual instances (e.g., images) which are linked by an abstract concept rather than any particular low-level visual similarity. Two instances of *Dog Walking* can be visually dissimilar, but conceptually analogous. While the notion of *situation* can applied to any abstract concept [6], most people would consider a visual situation category to be—like *Dog-Walking*—a named concept that invokes a collection of objects, regions, attributes, actions, and goals with particular spatial, temporal, and/or semantic relationships to one another.

The potential open-ended variation in components and relationships makes it difficult to model abstract situations, such as *Dog-Walking*, solely by learning statistics from a large collection of data. We believe that the process of analogy-making, as developed in [2], is a promising, though yet largely unexplored method for integrating prior conceptual-level knowledge with statistical learning in order to capture human-level flexibility in recognizing visual situations.

We are in the early stages of building a program, called *Situate*, whose goal is to flexibly categorize and explain new instances of known visual situations via analogy. In the long term, our system will integrate object-recognition and probabilistic-modeling methods with Hofstadter and Mitchell's *Copycat* architecture for analogy-making [7], [8]. Copycat's perceptual process interleaved top-down, expectation-driven actions with bottom-up exploration in order to recognize and make analogies between idealized situations. In Situate, we are building on Copycat's architecture to apply these ideas to visual perception and analogy-making.

In our envisioned system, Situate will attempt to make sense of a given visual image by interpreting it as an instance of a known situation prototype. To do this, Situate will locate objects, attributes, actions, and relationships relevant to the situation and, when possible, map these, with appropriate slippages, to the situation prototype. This mapping will allow Situate to (1) decide if the given image is an instance of the situation; and (2) if so, to *explain* how the system interpreted

M. Quinn is with the Department of Computer Science at Portland State University.

A. Rhodes is with the Department of Mathematics and Statistics at Portland State University.

M. Mitchell is with the Department of Computer Science at Portland State University and the Santa Fe Institute.

Fig. 1. Four instances of the "Dog-Walking" Situation. Images are from [1]. (All figures in this paper are best viewed in color.)

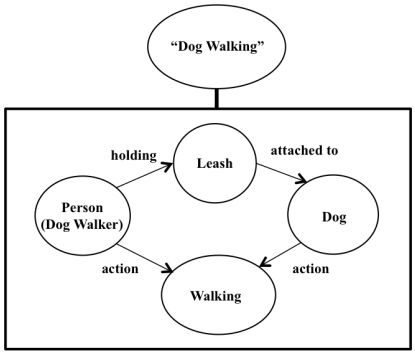

Fig. 2. A simple prototype for the Dog-Walking situation.

Fig. 3. Six "non-prototypical" *Dog-Walking* situations. (Images 1–4: [1]. Image 5: [4]. Image 6: [5].)

the image as such an instance. The explanation will explicitly indicate all the components of the mapping (including conceptual slippages, if needed) from the given image to the known prototype.

The problem of recognizing—and explaining—instances of a known visual situation shares motivation but contrasts with the widely studied tasks of "action recognition," "event recognition," or general "situation recognition" that have been explored in the computer vision literature (e.g., [9]–[13]). In such tasks, a system is asked to classify a new image as one of a set of possible action, event, or situation categories. While there has been significant recent progress on particular benchmark datasets, it is not surprising that such tasks remain very difficult for computer vision systems, which still perform well below humans. Moreover, even when such systems are successful in classifying images as instances of particular actions or events, it is not clear what such systems actually "understand" about those kinds of situations (though as a step in this direction, [10], [13] propose methods for assigning "semantic roles" in instances of action categories). As Figs. 1 and 3 illustrate, even the seemingly simple situation of "Dog-Walking" turns out to be complex and open-ended, requiring substantial background knowledge and the ability to deal with abstract concepts.

Thus, rather than attempting to classify scenes into one of many situation classes, our current work on Situate focuses on exhibiting deep knowledge of a *particular* situation concept. Our goal is a system that can, via analogy, fluidly map its knowledge of a particular abstract visual concept to a wide variety of novel instances, while being able to both explain its mapping and to measure how "stretched" that mapping is. We believe that this approach is the most likely one to capture general situation-recognition abilities. Moreover, we believe that the ability for such analogy-based explanations may also allow for new methods of training machine learning systems, such as indicating to a system not only that a classification was incorrect, but *why* it was incorrect.

Developing such a system is a long-term goal. In this paper we focus on a shorter-term subtask for Situate: using knowledge of situation structure in order to quickly locate objects relevant to a known situation. We hypothesize that using prior knowledge of a situation's expected structure, as well as situation-relevant context as it is dynamically perceived, will allow the system to be accurate and efficient at localizing relevant objects, even when training data is sparse, or when object localization is otherwise difficult due to image clutter or small, blurred, or occluded objects.

In the next section we describe this subtask in more detail. The subsequent sections give the details of our dataset and methods, results and discussion of experiments, an overview of related literature, and plans for future work.

II. Situation Structure and Efficient Object Localization

For humans, recognizing a visual situation is an active process that unfolds over time, in which prior knowledge interacts with visual information as it is perceived, in order to guide subsequent eye movements. This interaction enables a human viewer to very quickly locate relevant aspects of the situation [14]–[18].

Most object-localization algorithms in computer vision do not exploit such prior knowledge or dynamic perception of context. The current state-of-the-art methods employ feedforward deep networks that produce and test a fixed number of *object proposals* (also called *region proposals*)—on the order of hundreds to thousands—in a given image (e.g., [19]–[21]). An *object proposal* is a region or bounding box in the image (sometimes associated with a particular object class).

Assuming an object proposal defines a bounding box, the proposal is said to be a successful localization (or detection) if the bounding box sufficiently overlaps a target object's ground-truth bounding box. Overlap is measured via the intersection over union (IOU) of the two bounding boxes, and the threshold for successful localization is typically set to 0.5 [22].

Popular benchmark datasets for object-localization (or object-detection, which we will use synonymously, although they are sometimes defined distinctly) include Pascal VOC [22] and ILSVRC [23]. In each, the detection task is the following: given a test image, specify the bounding box of each object in the image that is an instance of one of M possible object categories. In the most recent Pascal VOC detection task, the number of object categories M is 20; in ILSVRC, M is 200. For both tasks, algorithms are typically rated on their *mean average precision* (mAP) on the task: the *average precision* for a given object category is the area under its precision-recall curve, and the mean of these values is taken over the M object categories. On both Pascal VOC and ILSVRC, the best algorithms to date have mAP in the range of about 0.5 to 0.70; in practice this means that they are quite good at locating some kinds of objects, and very poor at others.

In fact, state-of-the-art methods are still susceptible to several problems, including difficulty with cluttered images, small or obscured objects, and inevitable false positives resulting from large numbers of object-proposal classifications. Moreover, such methods require large training sets for learning, and potential scaling issues as the number of possible categories increases.

For these reasons, several groups have pursued the more human-like approach of "active object localization," in which a search for objects unfolds over time, with each subsequent time step using information gained in previous time steps (e.g., [24]–[26]).

Our approach is an example of active object localization, but in the context of specific situation recognition. Thus, only objects specifically relevant to the given situation are required to be located. Situate[1] is provided some prior knowledge—the set of the relevant object categories—and it learns (from training data) a representation of the expected spatial and semantic structure of the situation. This representation consists of a set of joint probability distributions linking aspects of the relevant objects. Then, when searching for the target objects in a new test image, the system samples object proposals from these distributions, conditioned on what has been discovered in previous steps. That is, during a search for relevant objects, evidence gathered during the search continually serves as context that influences the future direction of the search.

Our hypothesis is that this localization method, by combining prior knowledge with learned situation structure and active context-directed search, will require dramatically fewer object proposals than methods that do not use such information.

In the next sections we describe the dataset we used and experiments we performed to test this hypothesis.

[1]We refer to the system described in this paper as "Situate," though what we describe here is only one part of the envisioned Situate architecture.

III. Domain and Dataset

Our initial investigations have focused on the *Dog-Walking* situation category: not only is it easy to find instances to photograph, but, as illustrated by Figs. 1 and 3, it also presents many interesting challenges for the general tasks of recognizing, explaining, and making analogies between visual situations. We believe that the methods developed in Situate are general and will extend to other situation categories.

We test our system on a new image dataset, called the "Portland State Dog-Walking Images" [1], created by our group. This dataset currently contains 700 photographs, taken in different locations by members of our research group. Each image is an instance of a "Dog-Walking" situation in a natural setting. (Figs. 1 and 3 give some examples from this dataset.) Our group has also hand-labeled the Dog-Walker(s), Dog(s), and Leash(es) in each photograph with tight bounding boxes and object category labels.[2].

For the purposes of this paper, we focus on a simplified subset of the *Dog-Walking* situation: photographs in which there is exactly one (human) dog-walker, one dog, one leash, and unlabeled "clutter" (such as non-dog-walking people, buildings, etc) as in Fig. 1. There are 500 such images in this subset.

IV. Overview of Our Method

Our system's task is to locate three objects—*Dog-Walker*, *Dog*, and *Leash*—in a test image using as few object proposals as possible. Here, an "object proposal" comprises an object category (e.g., *Dog*), coordinates of a bounding box center, and the bounding box's size and aspect ratio.

We use the standard criterion for successfully locating an object: an object proposal's intersection over union (IOU) with the target object's ground-truth bounding box must be greater than or equal to 0.5. Our main performance metric is the median number of object-proposal evaluations per image needed in order to locate all three objects.

In this section we give an overview of how our system works. In the following sections we describe the details of what the system learns, how it uses its knowledge to search for objects, and the results of experiments that demonstrate the effect of situation knowledge on the system's efficiency.

A. Knowledge Given to Situate

For this task, the only knowledge given to Situate is the set of relevant object categories {*Dog-Walker*, *Dog*, and *Leash*}, and the assumption that each image contains exactly one instance of each of those categories. In future work we will investigate less restricted situations, as well as the integration of richer information about situations, such as additional properties of constituent objects and more varied relationships between objects.

[2]The photographs and label files can be downloaded at the URL given in [1]. Note that our collection is similar to the Stanford 40 Actions [27] "Walking the Dog" category, but the photographs in our set are more numerous, varied, and have bounding box labels for each relevant object.

B. Knowledge Learned by Situate

From training data (photographs like those in Fig. 1), our system learns the following probability models:

1) **Bounding-Box Size and Shape Priors.** For each of the three object categories, Situate learns distributions over bounding-box size, which is given as the *area ratio* (box area / image area), and shape, which is given as the *aspect ratio* (box width / box height). These prior distributions are all independent of one another, and will be used to sample bounding boxes in a test image before any objects in that image have been localized. (Note that our system does not learn prior distributions over bounding-box *location*, since we do not want the system to model photographers' biases to put relevant objects near the center of the image.)
2) **Situation Model.** Situate's learned model of the *Dog-Walking* situation consists of a set of probability distributions modeling the joint locations of the three relevant objects, as well as the joint area-ratios and aspect-ratios of bounding boxes for the three objects. These distributions capture the expected relationships among the three objects with respect to location and size/shape of bounding boxes. When an object proposal is labeled a "detection" by the system, the situation model distributions are re-conditioned on all currently detected proposals in order to constrain the expected location, size, and shape of the other as-yet undetected objects.

Details of these two learning steps are given in Sec. VI.

C. Running Situate on a Test Image

Following the Copycat architecture [7], Situate's main data structure is the *Workspace*, which contains the input test image as well as any object proposals that have scored highly enough. Situate uses its learned prior distributions and its conditioned situation model to select and score object proposals in the Workspace, one at a time.

At any time during a run on a test image, each relevant object category (*Dog-Walker*, *Dog*, and *Leash*) is associated with a particular probability distribution over locations in the image, as well as probability distributions over aspect ratios and area ratios for bounding boxes. Situate initializes the location distribution for each object category to uniform over the image, and initializes each object category's area-ratio and aspect-ratio distribution to the learned prior distributions (Fig. 4).

At each time step in a run, Situate randomly chooses an object category that has not yet been localized, and samples from that category's current probability distributions over location, aspect ratio, and area ratio in order to produce a new object proposal. (If part of the object proposal's bounding box lies outside of the image boundaries, the bounding box is cropped to the part that lies inside the image boundaries.) The resulting proposal is then given a score for that object category, as described in Sec. IV-D.

The system has two user-defined thresholds: a *provisional detection threshold* and a *final detection threshold*. These thresholds are used to determine which object proposals are promising enough to be added to the Workspace, and thus influence the system's subsequent search.

If an object proposal's score is greater than or equal to the *final detection threshold*, the system marks the object proposal as "final," adds the proposal to the Workspace, and stops searching for that object category.

Otherwise, if an object proposal's score is greater than or equal to the *provisional detection threshold*, it is marked as "provisional." If its score is greater than any provisional proposal for this object category already in the Workspace, it replaces that earlier proposal in the Workspace. The system will continue searching for better proposals for this object category.

Whenever the Workspace is updated with a new object proposal, the system modifies the current situation model to be conditioned on all of the object proposals currently in the Workspace.

The conditioned location distributions reflect where the system should *expect* to see instances of the associated object categories, given the detections so far. Similarly, the conditioned area-ratio and aspect-ratio distributions reflect what size and shape boxes the system should expect for the associated object categories, given what has already been detected.

The purpose of *provisional* detections in our system is to use information the system has discovered even if the system is not yet confident that the information is correct or complete. In Sec. VIII we describe the results of an experiment that tests the usefulness of provisional detections for directing the system's search for its target objects.

D. Scoring Object Proposals

In the experiments reported here, during a run of Situate, each object proposal is scored by an "oracle" that returns the intersection over union (IOU) of the object proposal with the ground-truth bounding box for the target object. The *provisional* and *final* detection thresholds are applied to this score to determine if the proposal should be added to the Workspace. For the experiments described in this paper, we used a provisional detection threshold of 0.25 and a final detection threshold of 0.5. This oracle can be thought of as an idealized "classifier" whose scores reflect the amount of partial localization of a target object.

Why do we use this idealized oracle rather than an actual object classifier? The goal of this paper is not to determine the quality of any particular object classifier, but to assess the benefit of using prior situation knowledge and active context-directed search on the efficiency of locating relevant objects. Thus, in this study, we do not use trained object classifiers to score object proposals.

In future work we will experiment with object classifiers that can predict not only on the object category of a proposal but also the amount and type of overlap with ground truth (e.g., see [28] for interesting work on this topic).

E. Main Loop of Situate

The following describes the main loop of Situate, in which the system searches for objects in a new test image.

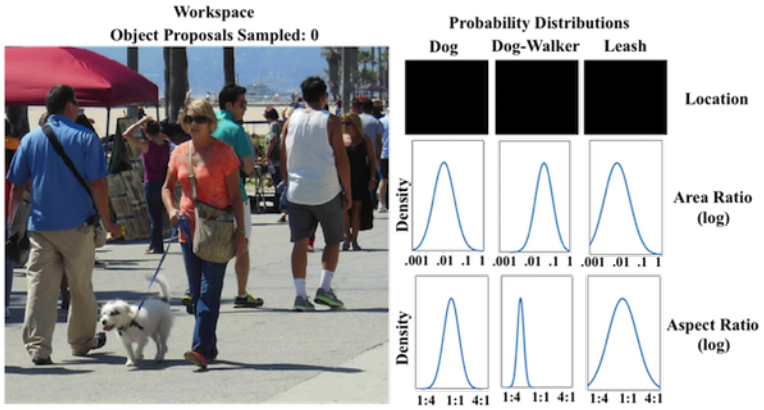

Fig. 4. Situate's initialization with a test image, before the start of a run. The Workspace is shown on the left, initialized with a test image. On the right, the initial per-category probability distributions for location, area ratio, and aspect ratio are shown. The (two-dimensional) location distributions for each object category are initialized as uniform (here, all black), and the area-ratio and aspect-ratio distributions are initialized to the learned prior distributions. In this visualization, the density axis has arbitrary units to fit the plot size; the intention here is to show the general shape of these distributions. For example, as expected, human dog-walkers tend to have higher area ratios and lower aspect ratios than dogs, and leashes have considerably more variation than either humans or dogs.

Input: A test image

Initialization: Initialize location, area-ratio, and aspect-ratio distributions for each relevant object category (*Dog-Walker*, *Dog*, and *Leash*). The initial location distributions are uniform; initial area-ratio and aspect-ratio distributions are learned from training data.

Main Loop: Repeat for *max-num-iterations* or until all three objects are detected:

1) Choose object category c at random from non-detected categories in {*Dog-Walker*, *Dog*, and *Leash*}.
2) Sample from category-specific location, area-ratio, and aspect-ratio distributions for category c to create an object proposal.
3) Calculate detection score d for this object proposal.
4) If $d \geq$ *final-detection-threshold*, mark proposal as "final," and add it to the Workspace.
5) Else, if $d \geq$ *provisional-detection-threshold*, mark proposal as "provisional. If d is greater than the detection score of any previous provisional proposal in the Workspace, add new proposal to the Workspace, replacing any previous provisional proposal for category c.
6) Update probability distributions in situation model to be conditioned on current proposals in the Workspace.

Return: number of iterations needed to successfully detect all three objects (a *completed situation detection*), or, if not successful, "failure".

In our experiments we set *max-num-iterations* per image to 1,000.

V. A Sample Run of Situate

In this section we illustrate this process with a few visualizations from a run of Situate—the same run whose initial state is shown in Fig 4. Fig. 5 shows the state of the system after it has iteratively sampled six object proposals. The first five scored below the provisional detection threshold, but the sixth proposal—for *Dog-Walker*—has a score (0.36) that is above the provisional detection threshold. Thus a provisional detection (dashed red box) is added to the Workspace. This causes the system to modify the current location, area, and aspect ratio distributions for the other two object categories, so that they are conditioned on this *Dog-Walker* proposal according to the learned situation model (right side of Fig 5). The new, conditioned distributions indicate where, and at what scale, the system should *expect* to locate the dog and the leash, given the (provisionally) detected dog-walker. The detected proposal does not affect the distributions for *Dog-Walker*; in our current system, an object category is not conditioned on itself.

The modified probability distributions for *Dog* and *Leash* allow the system to focus its search on more likely locations and box sizes and shapes. Fig. 6 shows the state of the system after 19 object proposals have been sampled. The system has successfully located the dog (the IOU with ground truth is 0.55). This final detection is added to the Workspace, and the system will no longer search for dogs. There are now two detections in the workspace. The *Leash* probability distributions are conditioned on both of them, and the *Dog* and *Dog-Walker* probability distributions are conditioned on each other (right side of Fig. 6).

Fig. 7 shows the system's state after 26 object proposals have been sampled. The updated situation model allows a better (and final) *Dog-Walker* proposal to be found quickly, as well as a final *Leash* proposal.

Taken together, Figs. 4–7 give the flavor of how Situate is able to use learned situation structure and active object localization in order to very quickly close in on the relevant objects in a visual situation.

VI. Situate's Probabilistic Models: Details

In this section we give details of how our system learns the probabilistic models of situations it uses to quickly locate relevant objects.

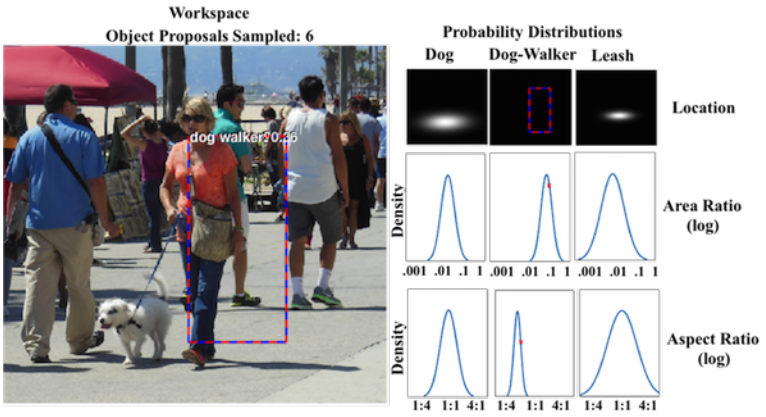

Fig. 5. The system's state after six object proposals have been sampled and scored. The sixth proposal was for the *Dog-Walker* category and its score (0.36) is higher than the provisional threshold, so a provisional detection was added to the Workspace (red dashed box; the samples that gave rise to this proposal are shown in red on the various *Dog-Walker* probability distributions). This causes the location, area ratio, and aspect ratio distributions for *Dog* and *Leash* to be conditioned on the provisional *Dog-Walker* detection, based on the learned situation model. In the location distributions, white areas denote higher probability. The area-ratio and aspect-ratio distributions for Dog and Leash have also been modified from the initial ones, though the changes are not obvious due to our simple visualization.

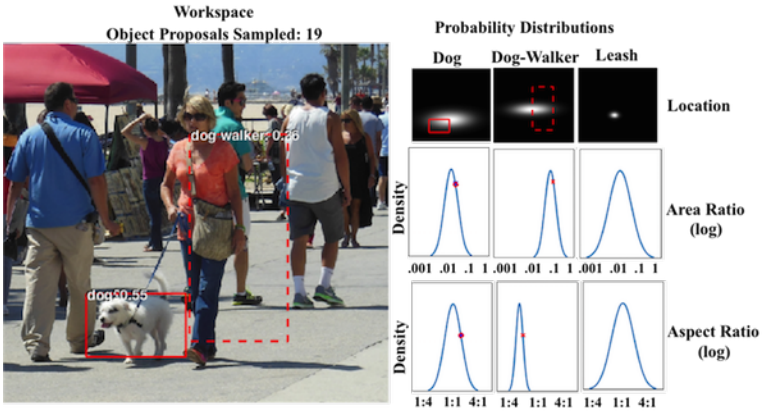

Fig. 6. The system's state after 19 object proposals have been sampled and scored. The focused conditional distributions have led to a *Dog* detection at IOU 0.55 (solid red box, indicating final detection), which in turn modifies the distributions for *Dog-Walker* and *Leash*. The situation model is now conditioned on the two detections in the Workspace. Note how strongly the *Dog* and *Dog-Walker* detections constrain the *Leash* location distributions.

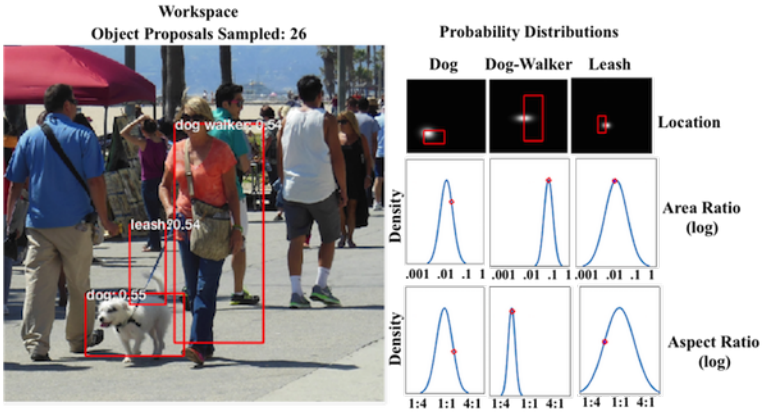

Fig. 7. The system's state after 26 object proposals have been sampled and scored. Final detections have been made for all three objects.

Before learning, all images in the dataset are scaled to have ∼250,000 pixels (preserving each image's original aspect ratio). The center of the image is assigned coordinates (0,0). The coordinates of all ground-truth boxes are converted to the new scale and coordinate system.

A. Bounding Box Size and Shape Priors

For each of the three object categories, Situate takes the ground-truth bounding boxes from the training set, and fits the natural logarithms of the box sizes (area ratio) and box shapes (aspect ratio) to normal distributions. At test time, the system uses these *prior* distributions to sample area ratio and aspect ratio until one or more detections have been added to the Workspace.

We used log-normal distributions to model these values rather than normal distributions, because the former are always positive and give more weight to smaller values. This made log-normal distributions a better fit for the data.

B. Situation Model

From training data, Situate learns a *situation model*: a set of joint probability distributions that capture the "situational" correlations among relevant objects with respect to location, area, and aspect ratio. As described above, when running on a test image, Situate will use these distributions in order to compute category-specific location, area, and aspect ratio probabilities *conditioned* on objects that have been detected in the Workspace.

The joint probability distributions Situate learns are the following:

Location distributions: Let $(x_{\text{Dog}}, y_{\text{Dog}})$, $(x_{\text{DW}}, y_{\text{DW}})$, $(x_{\text{L}}, y_{\text{L}})$ denote the coordinates of the bounding-box center for a Dog, Dog-Walker, and Leash, respectively.

Situate learns the pairwise and three-way location relationships among relevant objects, by modeling them as the following multivariate normal distributions:

$$(x_{\text{Dog}}, y_{\text{Dog}}, x_{\text{DW}}, y_{\text{DW}}) \sim \mathcal{N}(\boldsymbol{\mu}_{\text{Dog,DW}}, \boldsymbol{\Sigma}_{\text{Dog,DW}})$$
$$(x_{\text{Dog}}, y_{\text{Dog}}, x_{\text{L}}, y_{\text{L}}) \sim \mathcal{N}(\boldsymbol{\mu}_{\text{Dog,L}}, \boldsymbol{\Sigma}_{\text{Dog,L}})$$
$$(x_{\text{DW}}, y_{\text{DW}}, x_{\text{L}}, y_{\text{L}}) \sim \mathcal{N}(\boldsymbol{\mu}_{\text{DW,L}}, \boldsymbol{\Sigma}_{\text{DW,L}})$$
$$(x_{\text{Dog}}, y_{\text{Dog}}, x_{\text{DW}}, y_{\text{DW}}, x_{\text{L}}, y_{\text{L}}) \sim \mathcal{N}(\boldsymbol{\mu}_{\text{Dog,DW,L}}, \boldsymbol{\Sigma}_{\text{Dog,DW,L}}).$$

Here $\boldsymbol{\mu}$ is the multivariate mean over locations and $\boldsymbol{\Sigma}$ is the covariance matrix; these parameterize the distribution.

Size and Shape Distributions: Let α_{Dog}, α_{DW}, and α_{L} denote the natural logarithm of the bounding box area-ratio for a Dog, Dog-Walker, and Leash, respectively. Similarly, let γ_{Dog}, γ_{DW}, and γ_{L} denote the natural logarithm of the bounding box aspect-ratio for a Dog, Dog-Walker, and Leash, respectively.

Situate learns the pairwise and three-way size and shape relationships among relevant objects by modeling them as the following multivariate log-normal distributions over area ratio and aspect ratio:

$$(\alpha_{\text{Dog}}, \gamma_{\text{Dog}}, \alpha_{\text{DW}}, \gamma_{\text{DW}}) \sim \mathcal{N}(\boldsymbol{\mu}_{\text{Dog,DW}}, \boldsymbol{\Sigma}_{\text{Dog,DW}})$$
$$(\alpha_{\text{Dog}}, \gamma_{\text{Dog}}, \alpha_{\text{L}}, \gamma_{\text{L}}) \sim \mathcal{N}(\boldsymbol{\mu}_{\text{Dog,L}}, \boldsymbol{\Sigma}_{\text{Dog,L}})$$
$$(\alpha_{\text{DW}}, \gamma_{\text{DW}}, \alpha_{\text{L}}, \gamma_{\text{L}}) \sim \mathcal{N}(\boldsymbol{\mu}_{\text{DW,L}}, \boldsymbol{\Sigma}_{\text{DW,L}})$$
$$(\alpha_{\text{Dog}}, \gamma_{\text{Dog}}, \alpha_{\text{DW}}, \gamma_{\text{DW}}, \alpha_{\text{L}}, \gamma_{\text{L}}) \sim \mathcal{N}(\boldsymbol{\mu}_{\text{Dog,DW,L}}, \boldsymbol{\Sigma}_{\text{Dog,DW,L}}).$$

Here $\boldsymbol{\mu}$ is the multivariate mean over log-area-ratio and log-aspect-ratio and $\boldsymbol{\Sigma}$ is the covariance matrix.

We chose normal and log-normal distributions to use in the Situation Model because they are very fast both to learn during training and to condition on during a run on a test image. Their efficiency would also scale well if we were to add new attributes to the situation model (e.g., object orientation) and thus increase the dimensionality of the distribution.

However, such distributions sometimes do not capture the situation relationships very precisely. We discuss limitations of these modeling choices in Sec. X.

All of the code for this project was written in MATLAB, and will be released upon publication of this paper.

VII. Comparison Methods

In Sec. II, we stated our hypothesis: by using prior (given and learned) knowledge of the structure of a situation and by employing an active context-directed search, Situate will require dramatically fewer object proposals to locate relevant objects than methods that do not use such information. In order to test this hypothesis, we compare Situate's performance with that of four baseline methods and two variations on Situate. These are described in this section.

A. Uniform Sampling

As the simplest baseline, we use uniform sampling of location, area ratio, and aspect ratio to form object proposals. Uniform sampling uses the same main loop of Situate (cf. Sec. IV-E) but keeps the location, area ratio, and aspect ratio distributions fixed throughout the search, as follows. Given an object category $c \in \{$*Dog-Walker*, *Dog*, *Leash*$\}$, an object proposal is formed by sampling the following values:

- *Location*: The coordinates (x, y) of the bounding-box center are sampled uniformly across the entire image, independent of c.
- *Area Ratio*: A value α is sampled uniformly in the range $[\ln .01, \ln .5]$; the area ratio is set to e^{α}. This means that the area ratio will be between 1% and 50% of the image size, with higher probability given to smaller values. This gives a reasonable fit to the distribution of area ratios (independent of c) seen in the training set.
- *Aspect Ratio*: A value γ is sampled uniformly in the range $[\ln(.25), \ln(4)]$; the area ratio is set to e^{γ}. This gives a reasonable fit to the distribution of aspect ratios (independent of c) seen in the training set.

B. Sampling from Learned Area-Ratio and Aspect-Ratio Distributions

A second baseline method samples location uniformly (as in the uniform method described above), but samples area ratio and aspect ratio from the prior (learned) per-category distributions (cf. Sec. VI-A), keeping all these distributions fixed throughout the search. This method tests the usefulness of learning prior log-normal distributions of box area and aspect ratio.

C. Location Prior: Salience

As we mentioned above, our model does not include a learned prior distribution on location, because we do not want to model the photographer bias which tends to put relevant objects in the center of a photo.

To test a baseline method with a location prior, we used a fast-to-compute salience algorithm, similar to the one proposed in [29] (and extended in [30]). In particular, when given an image, our system creates a category-independent location prior by computing a salience map on each test image, and then normalizing it to form a probability distribution over location—the *salience prior*. Each pixel is thus given a value $p \in [0, 1]$ representing the system's assessment of the probability that it is in the center of a foreground "object" (independent of object category).

This method uses the main loop of Situate, but throughout the search the location distributions are fixed to this salience prior; the bounding box size and shape distributions are fixed to the uniform distributions described in Sec. VII-A.

In more detail, following [29], our salience computation decomposes an image into a set of feature maps, each indicating the presence of a low-level visual feature at a particular scale, and then integrates the resulting feature maps into a single salience map. The low-level features are inspired by features known to be computed in the primary visual cortex [31] including local intensity, color contrast, and presence of edge orientations. Our method differs from that of [29] only in parameter settings that allow us to decompose an image into feature maps quickly, to avoid aliasing artifacts and bias, and to avoid unrepresented edge orientations and spatial frequencies. We found that much of the salience in the resulting maps is located on the edges of objects, which is reasonable. However, because our purpose is to select points near the center of mass of objects, we smooth the resulting salience map with a Gaussian kernel with standard deviation approximately 10% of the image width. A similar variation of [29] was recently shown to compare well with other state-of-the-art saliency methods [32].

D. Randomized Prim's Object Proposals

As a final baseline for comparison, we use the Randomized Prim's Algorithm [33], a category-independent object-proposal method. Given an image, this method first creates a superpixel segmentation, and then creates object proposals by repeatedly constructing partial spanning trees of the superpixels. The nodes of the partial spanning trees are individual superpixels, and the edges are weights based on superpixel similarity, measured along several dimensions (e.g., color, common border ratio, and size). The algorithm constructs a partial spanning tree by starting with a single randomly chosen node, and then choosing new nodes to add to the tree probabilistically based on edge weight. A randomized stopping criterion is used to terminate a partial spanning tree. The final object proposal bounding-box is constructed to surround the superpixels in the tree. This procedure is repeated N times, where the number N of object proposals requested per image is a user-defined parameter. The number of proposals generated is not always equal to N since near-duplicate proposals are removed from the set of generated proposals. Note that the proposals produced are not ranked by the algorithm in any way. This algorithm was found to be competitive with several other 2014 state-of-the-art object proposal methods on Pascal VOC data [34].

To compare this method with Situate, we used the MATLAB implementation provided at https://github.com/smanenfr/rp. We used the value 200 for the "minimum number of superpixels," and $N = 10,000$. In order to get a large-enough set of bounding boxes, we ran the algorithm four times on each image, each time using a different color space (HSV, LAb, Opponent, and rg) for the color similarity feature (see [33] for details).

For each image in the test set, we randomly sampled 1,000 object proposals, one at a time (without replacement) from the generated set. If a sampled proposal bounding box had IOU ≥ 0.5 for any of the three relevant ground truth objects, that object was marked as "final," indicating a final detection. If all three objects were detected (marked as "final") during the sampling procedure, sampling was stopped and the number of proposals sampled was returned. Otherwise "failure" was returned after 1,000 proposals were sampled.

E. Combining Salience with Learned Distributions

In addition to these baselines, we also investigated two variations on our method. In the first, we experimented with a combination of methods, as follows. The prior location distributions for all categories are based on salience, and the prior distributions for bounding box size and shape are based on the learned log-normal distributions. After one or more detections are added to the Workspace, the conditional distributions associated with the situation model are used as described in previous sections, but with one change: the conditioned location distribution for each category is combined with the prior salience distribution by pointwise multiplication, followed by addition of a small non-zero value to each pixel (to avoid zero-probability locations), followed by renormalization to produce a new location distribution.

F. No Provisional Detections

In a second variation, we removed the system's ability to use provisional detections to condition situation model distributions. This means that the only kind of detections are final detections, which require $IOU \geq 0.5$. In this experimental condition, partial detections ($IOU \leq 0.5$) provide no information to the system. This condition tests the usefulness of using such partial information. We ran this using the "Combined Learned Distributions and Salience" version of our system, since that method exhibited the best performance, as we show in the next section.

VIII. Results

In this section we present results from running the methods described above. In reporting results, we use the term *completed situation detection* to refer to a run on an image

for which a method successfully located all three relevant objects within 1,000 iterations; we use the term *failed situation detection* to refer to a run on an image that did not result in a completed situation detection within 1,000 iterations.

The various methods described above are characterized by: (1) **Location Prior:** whether the prior distribution on location is uniform or based on salience; (2) **Box Prior:** whether the prior distributions on bounding-box size and shape are uniform or learned; and (3) **Situation Model:** whether, once one or more object detections are added to the Workspace, a learned situation model conditioned on those detections is used instead of the prior distributions, and whether that conditioned model is combined with a salience prior for location.

As described above, our dataset contains 500 images. For each method, we performed 10-fold cross-validation: at each fold, 450 images were used for training and 50 images for testing. Each fold used a different set of 50 test images. For each method we ran the algorithm described in Sec. IV-C on the test images, with *final-detection-threshold* set to 0.5, *provisional-detection-threshold* set to 0.25, and *maximum number of iterations* set to 1,000. In reporting the results, we combine results on the 50 test images from each of the 10 folds and report statistics over the total set of 500 test images.

A. Number of Iterations Per Image to Reach Completed Situation Detection

Fig. 8 gives, for each method, the median number of iterations per image in order to reach a completed situation detection. The medians are over the union of test images from all 10 folds—that is for 500 images total. The median value is given as "failure" for methods on which a majority of test image runs resulted in failed situation detections. We used the median instead of the mean to allow us to give a statistic that includes the "failure" runs.

Fig. 8 shows the strong benefit of using a situation model. The most effective method was the combination of a salience prior, learned box priors, and a learned situation model. For a majority of test images this method required fewer than 200 iterations to locate all three objects. The same method without salience performed only slightly worse, indicating that the salience prior had only a minor effect on the system's efficiency. It is also clear that using provisional detections to guide search is very helpful; removing this ability (the "No provis." case) nearly quadrupled the median number of iterations needed. The four comparison methods with no situation model failed to reach a completed situation detection on a majority of the test images.

A more detailed way to visualize results from our runs is given in Fig. 9. This plot shows, for each method, a cumulative count of completed situation detections as a function of number of iterations. In particular, it shows, for each number n of iterations, how many of the 500 test images had complete situation detections within n or fewer iterations. For example, for the best-performing method ("Salience, Learned, Learned"), about 250 test images had complete situation detections within 200 or fewer iterations. The two top methods—those using learned situation models that were influenced by provisional detections—both show a steep rise in the initial part of the curve, which means that for most of the test images only a small number of object proposal evaluations were needed.

Fig. 9 shows that even the best method failed to reach complete situation detections on about 75 of the 500 test images. By looking at the final state of the Workspace on these images, we found that the most common problem was locating leashes. In many of the failure cases, a partial leash was detected, but the system was unable to improve on the partial detection.

B. Sequence of Detections

The previous plots showed results for completed situation detections. What can we say about the sequence of object detections within each image? For methods using a situation model, we would expect that detecting the first object would take the most iterations. This is because, before a first object is detected, the system relies only on its prior distributions. But once a first object is detected, the situation model, conditioned on that detection, serves to constrain the search for a second object, and it should be detected much more quickly than the first object. Once a second object is detected, the situation model is conditioned on two detections, constraining the search even more, so we might expect the third detection to occur even more quickly.

We tested these expectations by recording, for each test image, the number of iterations to the first detection, from the first to the second, and from the second to the third. We denote these as $t_{0,1}$, $t_{1,2}$, and $t_{2,3}$, respectively. Here, by "detection," we refer to detections at the *final-detection-threshold*. Each method's median values of $t_{0,1}$, $t_{1,2}$, and $t_{2,3}$ over the 500 test images are shown in Fig. 10. As before, if a method fails to achieve a first, second, or third detection within 1,000 iterations, that detection is marked as "failure". In Fig. 10, a bar marked as "Failure" indicates that the method failed on a majority of the test images. (Results for Randomized Prims are not included here because we did not collect this finer-grained data for that method.)

Let $\tilde{t}_{0,1}$ denote the median time to the first detection, $\tilde{t}_{1,2}$ denote the median time from the first to the second detection, and $\tilde{t}_{2,3}$ denote the median time from the second to the third detection, for a given method. For the three methods using a situation model, Fig. 10 shows that, as expected, $\tilde{t}_{1,2}$ and $\tilde{t}_{2,3}$ are both considerably shorter than $\tilde{t}_{0,1}$. However, surprisingly, for the first two methods, $\tilde{t}_{2,3}$ is larger than $\tilde{t}_{1,2}$. This appears to be because dog-walkers (i.e., humans) are usually the first object to be detected (since they are the largest of the three objects and also the ones that best fit the learned probability distributions). dogs are usually detected second, and leashes usually detected last. Leashes tend to be the most difficult objects to detect, because they are in many cases quite small or otherwise are outliers with respect to the normal distributions learned from the training data. Conditioning on a learned situation model helps a great deal in locating these objects—after all, $\tilde{t}_{2,3}$ is relatively quite low—but the difficulty of detecting some leashes is most likely what causes $\tilde{t}_{2,3}$ to be larger than $\tilde{t}_{1,2}$ for the first two methods.

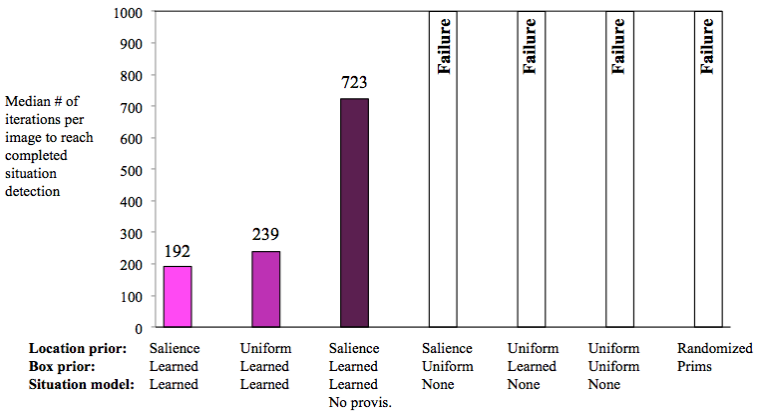

Fig. 8. Results from seven different methods, giving median number of iterations per image to reach a completed situation detection (i.e., all three objects are detected at final detected threshold). If a method failed to reach a completed situation detection within the maximum iterations on a majority of test images, its median is given as "Failure".

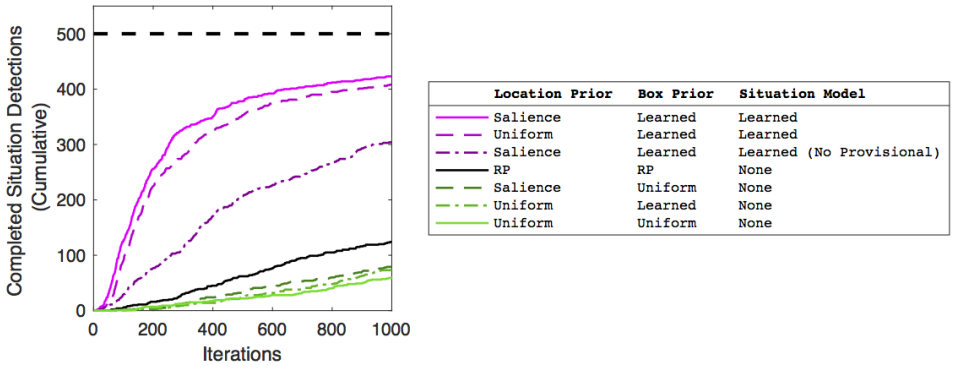

Fig. 9. Cumulative number of completed situation detections as a function of iterations. For each value n on the x-axis, the corresponding y-axis value gives the number of test image runs reaching completed situation detections with n or fewer object proposal evaluations. "RP" refers to the Randomized Prim's algorithm.

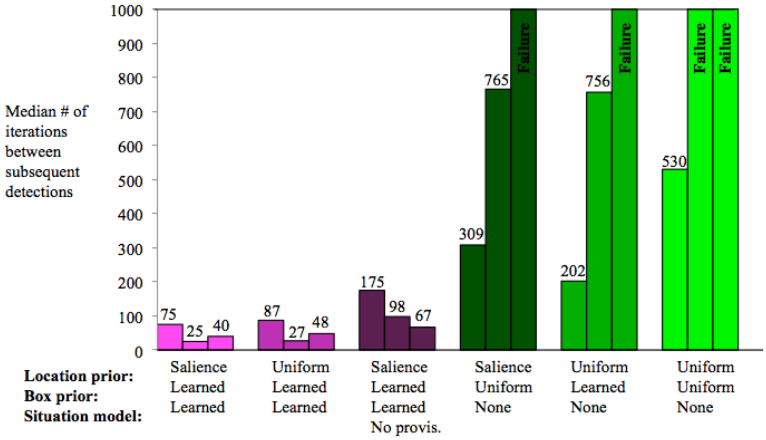

Fig. 10. Results from different methods giving median number of iterations between subsequent object detections (at the *final-detection threshold*. For each method, the plot gives three bars: the first bar is $\tilde{t}_{0,1}$, the median number of iterations to the first object detection in that image; the second bar is $\tilde{t}_{1,2}$, the median number of iterations from the first to the second detection; and the third bar is $\tilde{t}_{2,3}$, the median number of iterations from the second to the third detection. Results for Randomized Prims is not included here because we did not collect this finer-grained data for that method.

In the methods that do not use a situation model, $\tilde{t}_{1,2}$ and $\tilde{t}_{2,3}$ are much higher than $\tilde{t}_{0,1}$, which reflects the relative difficulty of detecting the three different object categories without the help of situational context.

C. Discussion

The results presented in this section strongly support our hypothesis that by using knowledge of a situation's structure in an active search, our method will require dramatically fewer

object proposals than methods that do not use such information. Situate's active search is directed by a set of probability models that are continually updated based on information gained by the system as it searches. Our results show that using information from provisional, incomplete detections is key to closing in quickly on a complete situation detection.

Our results showed that a location prior based on a fast-to-compute salience map only marginally improved the speed of localization. More sophisticated salience methods might reduce the number of iterations needed, but the computational expense of those methods themselves might offset the benefits. This is something we plan to explore in future work.

Since we used an idealized oracle to score object proposals, the results we report here support our hypothesis only if we assume that the system receives feedback about partial detections. In future work we will experiment with using a object classifier that is trained to provide feedback about the proposal's likely overlap with a target ground-truth region.

In addition, we found that, for the best-performing method, the failures to reach completed situation detections were largely due to the difficulty of locating leashes. This is an example of an object category that is very hard to locate without context, and even with strong contextual information it is often hard to go beyond a partial detection. To do this, it will be important to identify small object interactions, such as a person's hand holding a leash. Recent work in computer vision has addressed this kind of issue (e.g., [35]–[38]) and it will be one focus of our future research.

IX. Related Work

In this section we review work that is most closely related to our goals and methods: using context to localize objects, and active object localization.

A. Related Work on using Context to Localize Objects

Until recently the dominant paradigm in object localization and detection has been the exhaustive "sliding windows" approach (e.g., [39]). In the last several years, more selective approaches to creating category-independent object proposals have become popular (e.g., [40]). However, in order to make object detection more efficient and accurate, many groups have looked at how to use contextual features.

The term "context" takes on different meanings in different papers [41]. In some cases it refers simply to co-occurrence of objects, possibly in the same vicinity. For example, [39] uses co-occurrence statistics to re-score candidate bounding boxes ("sheep" and "cow" are more likely to co-occur in an image than "sheep" and "bus"). Several groups (e.g., [11], [42]–[45]) use graphical models to capture statistics related to co-occurrence well as some spatial relationships among objects in an image.

Other approaches use "Gist" and other global image features to relate overall scene statistics or categories to object locations, sometimes combining these features with co-occurrence statistics and spatial relationships between objects (e.g., [46]–[49]). Still other methods learn statistical relationships between objects and features of pixel regions that surround the objects (e.g., [50], [51]) or objects ("things") and homogeneous image regions ("stuff") [52]. Going further, [53] describes a method for learning joint distributions over scene categories (e.g., "street," "office," "kitchen") and context-specific appearance and layout of objects in those scenes.

More recently, several groups have combined object-proposal methods such as R-CNN [20], [40] with graphical models (e.g., [54]) or other CNN architectures (e.g., [10]) to combine object-to-object context with whole scene context in order to improve object detection.

The context-based methods described above are only a sample of the literature on incorporating context into object detection methods, but they give the general flavor of techniques that have been explored. Our work shares several goals with these approaches, but differs in at least four major aspects. One is our focus on images that are instances of abstract visual situations such as "Dog-Walking," in which "situational" context from relevant objects helps very significantly in object localization, as was shown in Sec. VIII. In contrast, the approaches described above have been designed and tested using image datasets in which the role of semantic or "situational" context is limited. This means that context of the kinds described above most often do not result in large improvements in localization accuracy [42].

A second aspect in which our work differs from other work on context is that we use prior situation knowledge to focus on specific object categories relevant to the given situation. The methods described above typically collect statistics on contextual relationships among *all* possible object categories (or at least all known to the system). The idea is that the system will learn which object categories have useful contextual relationships. This is an admirable but quite difficult task, and it entails some risks: with many categories and without enough training data, the system may learn false relationships among unrelated categories, and as the number of known categories increases, the amount of training data and computation needed for training and testing may scale in an untenable way. Our approach assumes that the system has some prior knowledge of a situation's relevant concepts; Situate is not meant to be a general statistical "situation-learning" system.

A third point of difference is the importance in our system of explanation to the user. As we described in Sec I, an key aspect of our envisioned system is its ability to explain its reasoning to humans via the structures it builds in the Workspace and their mappings to (and possible slippages from) a prototype situation. Creating an explicit situation model is part of the system's ability to explain itself. For example, visualizations like those in Figs. 5-7 make it very clear how the system's perception of context facilities object localization. In contrast, most other recent context-based object detection papers simply report on the difference context makes in per-category average precision—it is increased for some categories, decreased for others, but it is usually not clear why this happens, or what could be done to improve performance.

Finally, unlike our system, the approaches to context described above are not dynamic—that is, context is used on a test image as an added feature or variable used to optimize object detection. In our system, contextual features change

over time as a result of what the system perceives; there is temporal feedback between perception of context and object localization. This kind of feedback is the hallmark of active object localization, which leads to our discussion in the next subsection.

B. Related Work on Active Object Localization

Our method is an example of active object localization, an iterative process inspired by human vision, in which the evaluation of an object proposal at each time step provides information that the system can use to create a new object proposal at the next time step.

Work on active object localization has a long history in computer vision, often in the context of active perception in robots [55] and modeling visual attention [56]. The literature of this field is large and currently growing—here we summarize a few examples of recent work most similar to our own.

Alexe et al. [57] proposed an active, context-driven localization method: given an initial object proposal ("window") in a test image, at each time step the system uses a nearest-neighbor algorithm to find training image regions that similar in position and appearance to the current object proposal. These nearby training regions then "vote" on the next location to propose in the test image, given each training region's displacement vector relative to the ground-truth target object. These "vote-maps" are integrated from all past time steps; the integrated vote map is used to choose the next object proposal. The system returns the highest scoring of all the visited windows, using an object-classifier score. The authors found that their method increased mAP on the Pascal VOC 2010 dataset by a small amount over the state of the art method of the time, but used only about one-fourth the window evaluations as that method. However, the nearest-neighbor method can be costly in terms of efficiency. A more efficient and accurate version of this method, using R-CNN object proposals and random forest classifiers is described in [25].

Some groups have used recurrent neural networks (RNNs) to perform active localization. For example the work of Mnih et al. [58] (extended in [59]) combines a feedforward"glimpse network," which extracts a representation of an image region, with an RNN that inputs that representation as well as its own previous state to produce a new state that is input to an "action network" which outputs the next location to attend to. the algorithms described in [58] and [59] are tested on cluttered and translated MNIST digits as well as images of house numbers from Google Street View.

Several groups frame active object localization as a Markov decision process (MDP) and use reinforcement learning to learn a search policy. The approach proposed in [28] involves learning a search policy for a target object that consists of a sequence of bounding-box transformations, such as horizontal and vertical moves, and changes to scale and aspect ratio. The algorithm starts with a bounding box containing most of the image, and uses a deep reinforcement learning method to learn a policy that maps bounding box features to actions, obtaining a reward if IOU with the ground truth bounding box is reduced by the action. Once learned, this policy is applied to new test images to locate target objects. The authors tested this method on the Pascal VOC 2007 dataset; it obtained competitive mAP using on average a very small number of policy actions.

In the MDP method proposed in [60], an action consists of running a detector for a "context class" that is meant to help locate instances of the target "query class". To locate appropriate "context regions" in a test image, the system relies on stored pairs of bounding boxes from co-occurring object classes in training images, along with their displacement vector and change of aspect ratio. If the policy directs the system to detect a given context class in a test image, the system uses a nearest-neighbor method to create a vote map similar to that proposed in [57], as described above. This method was tested on the Pascal VOC 2010 dataset and exhibited a small increase in mAP over other state-of-the-art methods while requiring significantly fewer object-proposal evaluations

Nagaraja et al. [61] proposed an MDP method in which a search policy is learned via "imitation learning": in a given state, an oracle demonstrates the optimal action to take and the policy subsequently learns to imitate the oracle. The algorithm starts with a set of possible object proposals (generated by a separate algorithm) and its learned policy guides exploration of these proposals. The system was tested on a dataset of indoor scenes and was found to improve average precision (as a function of number of proposal evaluations) on several object categories as compared to a simple proposal-ranking method.

Like these methods, our approach focuses on perception as a temporal process in which information is used as it is gained to narrow the search for objects. However, the differences between our approach and these other active localization methods are similar to those we described in the previous subsection: often these methods are tested on datasets in which the role of context is limited; these methods often rely on exhaustive co-occurrence statistics among object categories; and it is usually hard to understand why these methods work well or fail on particular object categories. In addition, the reinforcement-learning based methods learn a policy that is fixed at test time; in our method, the representation of a situation itself adapts (via modifications to probability distributions) as information is obtained by the system. Finally, the amount of training data and computation time required can be quite high, especially for reinforcement learning and RNN-based methods.

In future work, we plan to compare some of these methods with ours on our situation-specific dataset, in order to assess the performance of these approaches on images in which context plays a large role, and to assess the relative requirements for training data and computation time among the different methods.

X. Conclusions and Future Work

Our work has provided the following contributions:

- We have proposed a new approach to actively localizing objects in visual situations, based on prior knowledge, adaptable probabilistic models, and information from provisional detections.
- We created a new situation-specific dataset (the Portland State Dog-Walking Images) with human-labeled object bounding boxes, and made it publicly available.

- We performed experiments comparing our system with several baselines and variations. These experiments demonstrated the benefits of our approach in the context of an idealized oracle classifier. We also analyzed where and why our approach fails.
- We contrasted our approach with those of several other research groups working on incorporating context into object detection, and on active object localization.

The work described in this paper is an early step in our broader research goal: to develop a system that integrates cognitive-level symbolic knowledge with lower-level vision in order to exhibit a deep understanding of specific visual situations. This is a long-term and open-ended project. In the near-term, we plan to improve our current system in several ways:

- Experimenting with probability models that better fit our target situations, rather than the more limited univariate and multivariate normal distributions used here. We are currently exploring versions of kernel density estimation and Gaussian process regression algorithms.
- Replacing the object-proposal scoring oracle with category-specific object classifiers, based on convolutional network features.
- Exploring more sophisticated salience methods to improve location priors, while taking into account the trade-off between the usefulness of the location prior and its computational expense.
- Creating datasets of other visual situations, and evaluating our approach on them. We would like to create a public "situation image dataset repository" for researchers interested in working on recognition of abstract visual situations. While there are numerous action- and event-recognition datasets available, we are not aware of any that are designed specifically to include a very wide variety of instances of specific abstract visual situations like those our method is aimed at.
- Comparing our system with related active object localization methods on situation-specific image datasets.

In the longer term, our goal is to extend Situate to incorporate important aspects of Hofstadter and Mitchell's Copycat architecture in order to give it the ability to quickly and flexibly recognize visual actions, object groupings, relationships, and to be able to make analogies (with appropriate conceptual slippages) between a given image and situation prototypes. In Copycat, the process of mapping one (idealized) situation to another was interleaved with the process of building up a representation of a situation—this interleaving was shown to be essential to the ability to create appropriate, and even creative analogies [8]. Our long-term goal is to build Situate into a system that bridges the levels of symbolic knowledge and low-level perception in order to more deeply understand visual situations—a core component of general intelligence.

ACKNOWLEDGMENTS

We are grateful to Jose Escario, Garrett Kenyon, Will Landecker, Sheng Lundquist, Clinton Olson, Efsun Sarioglu, Kendall Stewart, Mick Thomure, and Jordan Witte for discussions and assistance concerning this project. Many thanks to Li-Yun Wang for running the experiments with Randomized Prim's algorithm. This material is based upon work supported by the National Science Foundation under Grant Number IIS-1423651. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] "Portland State Dog-Walking Images." [Online]. Available: http://www.cs.pdx.edu/~mm/PortlandStateDogWalkingImages.html

[2] D. R. Hofstadter, *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. New York: Basic Books, 1995.

[3] ——, "Analogy as the core of cognition," in *The Analogical Mind: Perspectives from Cognitive Science*, D. Gentner, K. J. Holyoak, and B. N. Kokinov, Eds. MIT Press, 2001, pp. 499–538.

[4] "Dog Swimming." [Online]. Available: http://www.drollnation.com/gallery/2015/12/randomness-120315-5.jpg

[5] "Dog Walking Dog." [Online]. Available: http://www.delcopetcare.com/wp-content/uploads/2013/02/dog-walking.jpg

[6] D. Hofstadter and E. Sander, *Surfaces and Essences*. Basic Books, 2013.

[7] D. R. Hofstadter and M. Mitchell, "The Copycat project: A model of mental fluidity and analogy-making," in *Advances in Connectionist and Neural Computation Theory*, K. Holyoak and J. Barnden, Eds. Ablex Publishing Corporation, 1994, vol. 2, pp. 31–112.

[8] M. Mitchell, *Analogy-Making as Perception: A Computer Model*. MIT Press, 1993.

[9] G. Guo and A. Lai, "A survey on still image based human action recognition," *Pattern Recognition*, vol. 47, no. 10, pp. 3343–3361, 2014.

[10] S. Gupta and J. Malik, "Visual Semantic Role Labeling," *arXiv:1505.04474*, 2015.

[11] L.-J. Li and L. Li Fei-Fei, "What, where and who? Classifying events by scene and object recognition," in *International Conference on Computer Vision (ICCV)*. IEEE, 2007, pp. 1–8.

[12] L. Wang, Z. Zhe Wang, W. Wenbin Du, and Y. Qiao, "Object-scene convolutional neural networks for event recognition in images," in *Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. IEEE, 2015, pp. 30–35.

[13] A. Yatskar, M, Zettlemoyer, L., Farhadi, "Situation recognition: Visual semantic role labeling for image understanding," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[14] M. Bar, "Visual objects in context." *Nature Reviews Neuroscience*, vol. 5, no. 8, pp. 617–629, 2004.

[15] P. G. Malcolm, George L. and Schyns, "More than meets the eye: The active selection of diagnostic information across spatial locations and scales during scene categorization," in *Scene Vision: Making Sense of What We See*, M. Kveraga, K. and Bar, Ed. MIT Press, 2014, pp. 27–44.

[16] M. Neider and G. Zelinsky, "Scene context guides eye movements during visual search," *Vision Research*, vol. 46, no. 5, pp. 614–621, 2006.

[17] M. C. Potter, "Meaning in visual search," *Science*, vol. 187, pp. 965–966, 1975.

[18] C. Summerfield and T. Egner, "Expectation (and attention) in visual cognition." *Trends in Cognitive Sciences*, vol. 13, no. 9, pp. 403–9, 2009.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv:1512.03385*, 2015.

[20] R. Girshick, "Fast R-CNN," in *International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 1440–1448.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *arXiv:1506.02640*, 2015.

[22] M. Everingham, L. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[24] G. C. H. E. de Croon, E. O. Postma, and H. J. van den Herik, "Adaptive gaze control for object detection." *Cognitive Computation*, vol. 3, no. 1, pp. 264–278, 2011.

[25] A. Gonzalez-Garcia, A. Vezhnevets, and V. Ferrari, "An active search strategy for efficient object class detection," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 3022–3031.

[26] Y. Lu, T. Javidi, and S. Lazebnik, "Adaptive object detection using adjacency and zoom prediction," *arXiv:1512.07711*, 2015.

[27] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," in *International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 1331–1338.

[28] J. C. Caicedo and S. Lazebnik, "Active object localization with deep reinforcement learning," in *International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 2488–2496.

[29] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[30] L. Elazary and L. Itti, "A Bayesian model for efficient visual search and recognition," *Vision Research*, vol. 50, no. 14, pp. 1338–1352, 2010.

[31] L. Itti and C. Koch, "Computational modelling of visual attention," *Nature Reviews Neuroscience*, vol. 2, no. 3, pp. 194–203, 2001.

[32] S. Frintrop, T. Werner, and G. M. Garcia, "Traditional saliency reloaded: A good old model in new shape," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 82–90.

[33] S. Manen, M. Guillaumin, and L. V. Gool, "Prime object proposals with randomized Prim's algorithm," in *International Conference on Computer Vision (ICCV)*. IEEE, 2013, pp. 2536–2543.

[34] J. Hosang, R. Benenson, and B. Schiele, "How good are detection proposals, really?" *arXiv:1406.6962*, 2014.

[35] J. Oramas-M. and T. Tuytelaars, "Recovering hard-to-find object instances by sampling context-based object proposals," *arXiv:1511.01954*, 2015.

[36] A. Rosenfeld and S. Ullman, "Visual Concept Recognition and Localization via Iterative Introspection," *arXiv:1603.04186*, 2016.

[37] S. Singh, D. Hoiem, and D. Forsyth, "Learning to Localize Little Landmarks," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2016.

[38] B. Yao and L. Fei-Fei, "Modeling mutual context of object and human pose in human-object interaction activities," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 17–24.

[39] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–45, 2010.

[40] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014, pp. 580–587.

[41] C. Galleguillos and S. Belongie, "Context based object categorization: A critical survey," *Computer Vision and Image Understanding*, vol. 114, no. 6, pp. 712–722, 2010.

[42] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky, "Exploiting hierarchical context on a large database of object categories," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 129–136.

[43] X. Cao, X. Wei, Y. Han, and X. Chen, "An object-level high-order contextual descriptor based on semantic, spatial, and scale cues." *IEEE Transactions on Cybernetics*, vol. 45, no. 7, pp. 1327–39, 2015.

[44] S. Divvala and D. Hoiem, "An empirical study of context in object detection," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1271 – 1278, 2009.

[45] C. Galleguillos, "Object categorization using co-occurrence, location and appearance," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[46] S. Marat and L. Itti, "Influence of the amount of context learned for improving object classification when simultaneously learning object and contextual cues," *Visual Cognition*, vol. 20, no. 4-5, pp. 580–602, 2012.

[47] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014, pp. 891–898.

[48] C. Szegedy, S. Reed, D. Erhan, and D. Anguelov, "Scalable, high-quality object detection," *arXiv:1412.1441*, 2014.

[49] A. Torralba and K. Murphy, "Context-based vision system for place and object recognition," in *Ninth IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2003, pp. 273–280.

[50] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, "Multi-class segmentation with relative location prior," *International Journal of Computer Vision*, vol. 80, no. 3, pp. 300–316, 2008.

[51] Y. Zhu, R. Urtasun, R. Salakhutdinov, and S. Fidler, "SegDeepM: Exploiting segmentation and context in deep neural networks for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 07-12-June, 2015, pp. 4703–4711.

[52] G. Heitz and D. Koller, "Learning spatial context: Using stuff to find things," in *European Conference on Computer Vision (ECCV)*, 2008, pp. 30–43.

[53] H. Izadinia, F. Sadeghi, and A. Farhadi, "Incorporating scene context and object layout into appearance modeling," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014, pp. 232–239.

[54] W. Chu and D. Cai, "Deep feature based contextual model for object detection," *arxiv:1604.04048*, apr 2016.

[55] D. H. Ballard, "Animate vision," *Artificial Intelligence*, vol. 48, no. 1, pp. 57–86, 1991.

[56] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 185–207, 2013.

[57] B. Alexe, N. Heess, Y. Teh, and V. Ferrari, "Searching for objects driven by context," in *Advances in Neural Information Processing Systems*, 2012, pp. 1–9.

[58] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 2204–2212.

[59] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," *arXiv:1412.7755*, 2014.

[60] X. S. Chen, H. He, and L. S. Davis, "Object detection in 20 questions," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016, pp. 1–9.

[61] V. K. Nagaraja, V. I. Morariu, and L. S. Davis, "Searching for Objects using Structure in Indoor Scenes," *arXiv:1511.07710*, 2015.