Active Object Search by Learning Policies for View Aggregation and Selection

Sha Hu, Wang Zhu, Zhiwei Deng, Greg Mori

***Abstract*— We present an approach for active object search – exploring an environment to find every instance of an object of interest. We develop a novel strategy for view aggregation, integrating the information from multiple views of a scene to construct an accurate estimate of the locations where the objects of interest are present in the scene. This view aggregation strategy is deployed with a policy for determining view selection, which view to acquire next. The parameters of the view aggregation and view selection approaches are learned end-to-end within a unified framework. We ground this active object search within the task of guiding a UAV to find all the people present in a scene. A simulation environment is developed in order to learn parameters and validate the efficacy of the learned strategies. Empirical results in this environment demonstrate that effective policies can be learned that are superior to baseline methods.**

I. INTRODUCTION

Great strides have been made in visual recognition in recent years. State of the art learning approaches have demonstrated impressive performance at object recognition. Detecting/classifying objects in images captured by humans is arguably a solved problem, especially when these are common objects imaged in standard poses from canonical viewpoints, for which reams of data can be acquired from the internet.

However, the visual world is a complex place, with views and images that extend far beyond these canonical inputs. The problem of active object search, determining how best to move an active agent in the world to find objects of interest, extends the challenges of visual interpretation of input data. Not only must an agent process a larger variety of possible imagery, but the agent must also glean from these images where the objects of interest are and which views best for further refining an estimate of where these objects might be. This active vision problem is a classic one, with substantial previous literature (e.g. surveyed by [1]). Recent years have seen renewed attempts at this problem, bringing to bear modern deep learning approaches for interpreting input images and deep reinforcement learning models for planning policies.

In this paper we contribute to this vein of work. In particular, we develop a novel approach for learning how to perform view aggregation and view selection. We demonstrate this approach for the task of active object search, finding humans in a scene. This task is depicted in Fig. 1. An unmanned aerial vehicle (UAV) is searching for humans in a city scene. The UAV acquires images while moving around the environment. The goal is to efficiently and accurately detect all the humans in the scene. Meeting this goal involves (1) view aggregation: integrating information about human presence acquired by the UAV across different views; and (2) view selection: deciding where the UAV should move next in order to best acquire useful information about people in the scene.

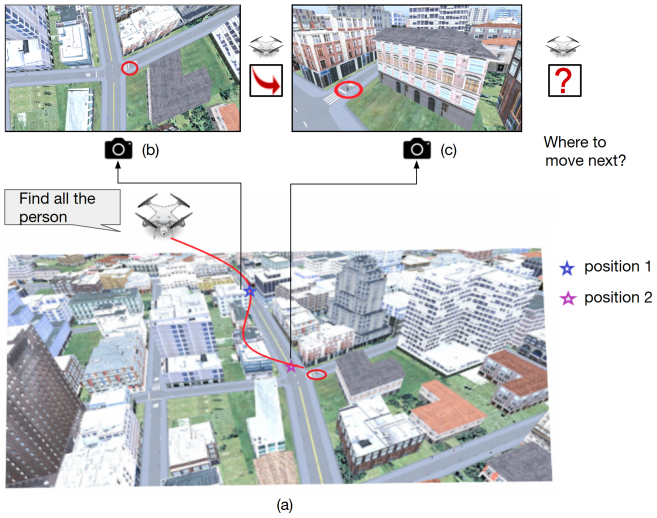


Fig. 1: Active object search by a UAV in a city scene. (a) Third-person illustration of the trajectory that the UAV follows. (b) Observation made by the UAV in position 1. In this viewpoint, the UAV can perceive a broad view of the scene. Some pedestrians in the streets are clearly distinguishable, while the person marked in the red circle is difficult to detect – it comprises only a few pixels and the visual evidence is ambiguous. (c) Observation made by the UAV in position 2. From this viewpoint, the person can be more easily detected.

The contribution of this paper are in developing a learning approach to address these two components. We propose a novel temporal view aggregation model that receives as input the images acquired by the UAV along with UAV pose. We cast an occupancy grid onto these images and the model performs object recognition to localize people in this occupancy grid. This occupancy grid representation is updated over time with a learned approach for aggregating the information from different views. Secondly, we demonstrate that this view aggregation model can be embedded within a reinforcement learning framework to learn a policy for

Authors are with the School of Computing Science, Simon Fraser University, Burnaby, BC Canada {hushah, wang_zhu, zhiweid, mori}@sfu.ca

view acquisition. This policy uses the current estimate for the occupancy map and visual input, along with the pose of the agent, to determine where to next move the UAV. We build a simulation environment in which UAV image inputs can be acquired across a variety of city scenes with different placements for the objects of interest (humans). This provides us with the data sources that can be used to train our proposed models. We demonstrate that our proposed approach can learn effective active object search strategies in this environment.

II. RELATED WORK

We develop a novel approach for active object search. There has been substantial previous work in this domain. In this section, we briefly review closely related approaches.

A. Occupancy Grids

We utilize an occupancy grid as our representation for object presence in the environment. Occupancy grids are widely used as a tool to represent the environment. An occupancy grid map is a 2d spatial map whose grid cells can for instance represent a probabilistic estimation of object presence at a discretized spatial location representation of the world.

Seminal work in this direction include [2], [3], [4]. This line of work focuses on multi-sensor fusion, and computes occupancy grid probabilities by combining the likelihood functions of different sensors using Bayesian statistics. These methods generally use hand-engineered approaches for determining likelihoods for each sensor and naive Bayes-type models for combination. In our work, we develop a novel deep learning based inverse sensor model and learned confidence for fusing information obtained from different camera views.

B. View Aggregation for View-Based Multi-View Recognition

Convolutional neural network (CNN) architectures have been extended to allow for recognition from image sequences using a single network. Su et al. [5] propose to use max-pool layers to combine information from multiple views of a 3D shape. These view are obtained by manipulating a camera over a fixed trajectory, which are combined into a single and compact shape descriptor via the pooling approach. Shi et al. [6] propose to wrap an object shape into a panorama and max pool across each resulting row. Both these methods assume a fixed-length image sequence is provided during both training and testing, thus they cannot generalize to the more general test setting where the view sequences are arbitrary and unconstrained.

Our confidence-based fusion for view aggregation can handle variable length view sequences, and we aim to fuse classification results of each view rather than fuse intermediate representations of each view. Further, we explore a multi-object active search setting. For our approach, different regions of one view can have different fusion weights in the temporal domain, leading to different aggregation across the different grid cells in our occupancy map representation.

C. View Planning for Active Vision

An active vision system is one that can manipulate the camera pose in the world in order to explore the environment and obtain better information from it [7], [8]. Recent work considers tasks such as active recognition [9], [10], [11], [12], [13], [14], active tracking [15], active detection [16], [17], active exploration [18], [19] and target driven visual navigation [20], [21]. We briefly review view planing for active object recognition.

Wu et al. [9] and Jayaraman and Grauman [11] compute information gain in next-best-view prediction for active categorization. Both methods learn to predict the next views of unseen test objects conditioned on various candidate agent motions starting from the current view, either by estimating 3D models from 2.5D RGBD images [9] or by learning to predict feature responses to camera motions [11]. They then estimate the information gain on their category beliefs from each such motion, and finally greedily select the estimated most informative next-best-view. Compared to [9] and [11], we focus on a multi-object active search task which requires holistic view aggregation across different spatial locations in the world.

Johns et al. [13] argues that the next-best-view should not only depend on the current view, but also a history of all past views. Malmir et al. [10] and Paletta and Pinz [22] use accumulated posterior belief over object label which is acquired by Naives Bayes to represent the state of the active object recognition system in each time step. Malmir and Cottrell [12] develop an efficient search procedure for active object recognition. Jayaraman and Grauman [14] modeled object exploration policy as a recurrent neural network and trained it using classification accuracy as reward. They found that predicting the next state of the environment based on current state and action improves the overall active object recognition accuracy.

D. Active Object Recognition Datasets

Current datasets that have been explored for active single object recognition include Wu et al. [9] and Malmir et al. [10]. ModelNet is a large collection of 3D CAD models for objects; however the resulting images lack complex backgrounds and occlusions. GERMS has 6 videos each of 136 objects being rotated around different fixed axes, against a television screen displaying moving indoor scenes. This results in challenging scenes, however the scale of the dataset is small and does not permit simulation for learning of active vision strategies. In this work, we propose a large synthetic dataset that can be useful for active object search for multiple objects in a single scene. The dataset includes complex backgrounds. The proposed dataset provides high-quality realistic 3D outdoor urban scenes with pedestrians on the streets, and integrated with Airsim [23] to allow training intelligent agents to navigate the 3D scenes.

III. PROBLEM STATEMENT AND APPROACH OVERVIEW

We address the problem of active object search. The application scenario is that of searching a given area to find all the instances of an object of interest. We specifically ground this in the task of a UAV equipped with a camera that is surveying a scene. The goal of the UAV is to detect all the humans in a designated area. From some viewpoints, the UAV can see some humans in the designated area clearly while other instances are not clearly distinguishable. Further, parts of the area will be imaged multiple times during the flight. Thus, the UAV needs to make intelligent movements and fuse information in order to obtain accurate understanding of how the objects of interest are distributed in this area as quickly as possible.

A. Our Task Formulation

We formulate the active object search as making probabilistic estimates on a human occupancy map as a process of simultaneous multiple measurement fusion and sequential decision making.

We first introduce the definition of the human occupancy map. Let W be a 3D world plane discretized into an N by N grid, y_i is a binary random variable that represents if there is person in the grid cell g_i. $p(y_i = 0)$ is the probability that there is no person in grid cell g_i, and $p(y_i = 1)$ is the probability there is at least one person in grid cell g_i. A probabilistic human occupancy map $M = [p(y_1), p(y_2), \ldots, p(y_{NN})]$ represents a belief over where the people are over the world plane W.

The agent observes the environment via a RGB camera, and has access to its pose. Let $\mathscr{A}$ be the set of all actions. Given a start pose and a time budget T, the agent is allowed to actively explore in the 3D world by taking a sequence of actions, utilizing the acquired RGB image sequence $O_{1:T} = \{O_1, O_2, \ldots, O_T\}$ and its pose sequence $P_{1:T} = \{P_1, P_2, \ldots, P_T\}$ to estimate the posterior probability over the occupancy map $M : P(y_1, \ldots, y_{NN}) \mid O_{1:T}, P_{1:T})$.

The performance of the agent is evaluated by the accuracy and F1 score, and we formally define the evaluation metric in Sec. VII-A.

B. Our Approach Overview

To solve this active occupancy map estimation problem, we propose an approach that can learn to fuse sequential classification results given by a classifier and learn a policy to directly optimize our goal of maximizing the performance of multiple grid cell classifications. The sequential fusion results will be taken as a part of state estimation for the policy and the policy will influence the input to the classifier and fusion module in the next time step by emitting an action to change the viewpoint of the agent. At the end of the time budget, the output of the fusion module will be taken as the final estimation of the map. We formulate this sequential fusion and decision making process in a reinforcement learning setup, and train the model with both standard backpropagation and REINFORCE. See Fig. 2 for a schematic showing how the modules are connected.

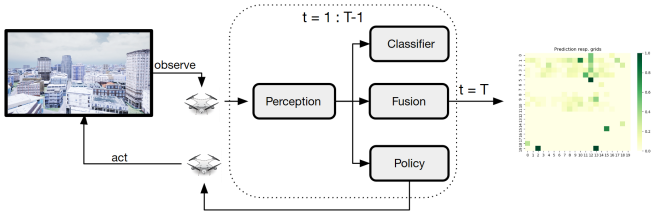


Fig. 2: A high-level illustration of the model depicting the interaction between perception, classifier, confidence-based measurement fusion and policy modules, unrolled over timesteps.

IV. MODEL ARCHITECTURE

Our model is composed of six basic modules. At each time t, the model receives a current view of the world O^t as input. We first map the positions of the grid cells from world coordinates to image coordinates and obtain a binary mask m_i^t that can present the pixels of the grid cell g_i in observation O^t (Sec. IV-A). Then the masked observation is fed to the feature extraction network to obtain the representation for the grid cells (Sec. IV-B). The grid feature is the input to both the classification network (Sec. IV-C) and the confidence prediction network (Sec. IV-D) to obtain the posterior estimation of whether the grid cell is occupied by a human and the confidence score for this estimation, respectively. With the estimations and scores, the fusion module (Sec. IV-E) can produce an aggregated estimation which will be encoded together with current observation and pose to the state in policy network (Sec. IV-F). Fig. 3 illustrates our network architecture.

In the following subsections we provide operational details of these components of our method.

A. World Frame to Image Frame Projection

To estimate if the grid cell g_i is occupied by a human, the agent should first determine where is the grid cell g_i in each observation. This can be completed through a forward camera projection.

Given current pose of the agent P_t and the 3D positions of the corners of the grid cell in world coordinates, we compute the positions of those corners in 2D pixel image coordinates. Each grid cell g_i is composed of four corners $\left\{p_i^{W_n}\right\}_{n=1,\ldots,4}$, and the 3D position of each corner is represented in homogeneous 3D world coordinates as $p_i^W = [x_i^W, y_i^W, 0, 1]^T$. We use a pinhole camera model [24] to project each p_i^W to the pixel coordinate represented as $p_i^C = [x_i^C, y_i^C, 1]^T$ in the image reference frame:

$$p_i^C = F_{intrinsic}(f_x, f_y, p_x, p_y) F_{extrinsic}(P_t) p_i^W \qquad (1)$$

where $F_{extrinsic}(\cdot)$ is a world-to-camera affine transformation function deterministically computed from the agent pose P_t, and $F_{intrinsic}(\cdot)$ is a camera-to-image transformation function

computed from the focal lengths of the camera f_x and f_y, and the principal point $[p_x, p_y]^T$.

At time t, we use a binary-valued mask m_i^t to indicate which pixels in observation O^t are within the area of grid cell g_i. Pixels within the convex shape that are composed by the pixels $\left\{p_i^{C_n}\right\}_{n=1,...,4}$ are valued 1 and pixels outside the convex shape are valued 0.

B. Grid Perception Network

The grid perception network $F_{perception}$, parameterized by θ_f, is to encode the observation into a feature vector f_i^t and provides this as input to the classification network and the confidence prediction network.

$$f_i^t = F_{perception}(O^t \odot m_i^t; \theta_f) \tag{2}$$

where $\odot$ is element-wise multiplication.

In our experiments, we extract fc7 features from a fine-tuned VGG-19 network [25] and use $f_i^t \in \mathbb{R}^{4096}$.

C. Grid Classification Network

The classification network $F_{classifier}$, parameterized by θ_c, is to classify if there is person in grid cell g_i given feature f_i. We use three fully connected layers with each followed by a ReLU activation function and a dropout layer, and a log-softmax layer:

$$l_i^t = F_{classifier}(f_i^t; \theta_c) \tag{3}$$

where l_i^t is a 2-dimensional vector representing the likelihood of the grid cell g_i being occupied by at least one person measured by the observation at time step t, i.e., $l_i^t = [p(y_i^t = 0), p(y_i^t = 1)]$.

D. Confidence Prediction Network

We are interested in inferring the uncertainty of the output of the neural networks from the input, and we propose to train a network to predict this uncertainty. The goal of the confidence prediction network is to estimate how confident the classifier network is given the input visual feature f_i^t. Instead of emitting a score that is representative of the failure probability of the system as in [26] where the score itself will not be used, the uncertainty of the classifier, denoted as w_i^t, will be adopted to weight each measurement later in the measurements fusion module. We propose that the confidence score can be inferred from the visual feature space directly:

$$w_i^t = F_{confidence}(f_i^t; \theta_w) \tag{4}$$

where $F_{confidence}(\cdot)$ is three fully connected layers with each followed by ReLU activation function and a dropout layer, parameterized by θ_w.

The ideal confidence prediction network can learn to output high scores for visual features that lead to correct classification of the classification network, and output low scores to visual features that lead to incorrect classification. This property can lead to a high performance of the final aggregated predictions.

E. Multiple Measurements Fusion with Learned Confidence

The purpose of the fusion module is to generate a prediction for a grid cell g_i given predictions from observations taken from a sequence of different viewpoints.

The final binary classification result for grid cell g_i, denoted as $\overline{l_i}$, is computed from T measurements $\left\{l_i^1, l_i^2, ..., l_i^T\right\}$ and corresponding confidence scores of each measurement $\left\{w_i^1, w_i^2, ..., w_i^T\right\}$. We normalize the predicted confidence scores by applying a $Softmax$ across T confidence scores. The normalized confidence scores, denoted as $\left\{\hat{w}_i^1, \hat{w}_i^2, ..., \hat{w}_i^T\right\}$, can then be used to aggregate the T measurements:

$$\hat{w}_i^t = \frac{e^{s_i^t}}{\sum_{\tau=1}^{T} e^{s_i^\tau}} \tag{5}$$

$$\overline{p}(y_i = 0) = \sum_{t=1}^{T} \hat{w}_i^t \times p\left(y_i^t = 0\right) \tag{6}$$

$$\overline{l_i} = [\overline{p}(y_i = 0), \overline{p}(y_i = 1)] \tag{7}$$

where $\overline{p}(y_i = 1)$ is the aggregated probability indicates the probability that grid cell g_i is occupied by at least one human.

F. Policy Network

The goal of the policy network is to output a sequence of actions that leads to accurate occupancy map estimation results. The policy network has a non-standard neural net architecture involving stochastic units: at each time step, it outputs a probability distribution $p(a \mid s^t)$ over the candidate action space $\mathscr{A}$ given state s^t, from which it use Monte Carlo sampling to sample one action a^t. In our experiment setting, The available actions for the agent are predefined as 4 discrete actions, namely $\mathscr{A} = \{MoveAhead, MoveRight, MoveLeft, MoveBack\}$ with the camera orientation fixed. We hypothesize that modeling an incrementally built belief about the environment can help the agent to make decisions. Thus, the incrementally built occupancy map is fed to the state encoding function together with the current observation and position.

$$p(a \mid s^t) = F_{action}(s^t; \theta_a) \tag{8}$$

$$s^t = F_{state}(F_{embedding}(O^t; \theta_e) \oplus c^t \oplus \overline{M^t}; \theta_{st}) \tag{9}$$

where c^t is a binary-valued 2D tensor of the same size as the occupancy map to indicate which cell is within the current field of view O^t. $\overline{M^t}$ is the estimated occupancy map at time t with fused measurements from $1:t$. $F_{action}(\cdot)$ is two fully connected layers with each followed by ReLU activation and a softmax layer. $F_{embedding}(\cdot)$ has the same architecture as F_{vgg} with one fully connect layer added to project the 4096 dimensional features to 512 dimensional embeddings. $F_{state}(\cdot)$ is two fully connected layers with each followed by ReLU activation.

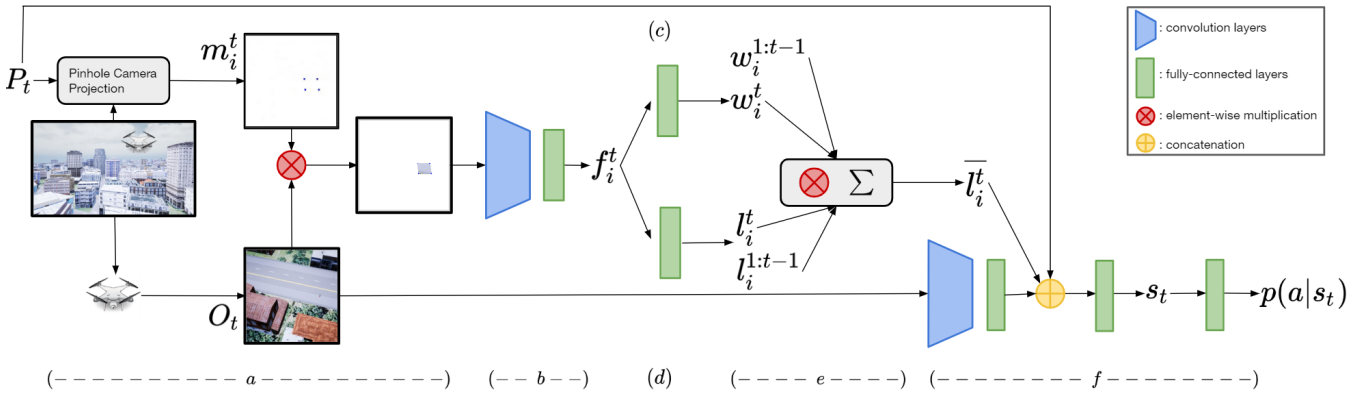


Fig. 3: Illustration of our model structure. (a) Pinhole camera projection module. (b) Grid perception network. (c) Confidence prediction network. (d) Grid classification network. (e) Measurements fusion module. (f) Policy network.

V. Training

We use two stages of training to learn the weights of our model $\Theta = [\theta_f, \theta_c, \theta_w, \theta_a, \theta_e, \theta_{st}]$.

In the first stage (Sec. V-A), we aim to train a standard neural network based visual feature extraction and classification network in a one view setting. We then freeze the parameters of the perception and classifier network and provide the property of the two parts in Sec. VII-C.

Since the policy network contains non-differentiable components, the parameters of the policy network are trained using REINFORCE [27]. The performance of the confidence prediction network is evaluated at the end of a sequence of observations, and how a sequence of observations is obtained is dependent on the policy network, Therefore, we design the second stage of training (Sec. V-B) in a multi-view setting where the confidence prediction network and the policy network are trained in a fully end-to-end fashion using a combination of backpropagation and REINFORCE.

A. One view training

We use the softmax loss as the objective and standard backpropagation to train the perception and classification network in a one view setting. In this one view setting, T is set to 1. Then:

$$[\Delta\theta_f, \Delta\theta_c] = -\sum_{n=0}^{N} \nabla_{\theta_f,\theta_c} L_{softmax}(l_n, y_n) \tag{10}$$

where l_n is the one view grid classification result directly from the grid classifier as in Eqn. 3, and y_n is the ground truth label for g_i. N is the number of observations obtained from random trajectories.

B. Multi-view training

We also use the softmax loss to train the confidence prediction network:

$$[\Delta\theta_w] = -\sum_{n=0}^{N} \nabla_{\theta_w} L_{softmax}(\overline{l_n}, y_n) \tag{11}$$

where $\overline{l_n}$ is the fused classification output from the multiple measurements fusion module as in Eqn. 7. N is the number of episodes.

The REINFORCE gradient is computed using the approximation proposed in [27]:

$$[\Delta\theta_a, \Delta\theta_e, \Delta\theta_{st}] \approx \sum_{i=1}^{N}\sum_{t=1}^{T} \nabla_{\theta_a,\theta_e,\theta_{st}} \log p(a_i^t \mid s_i^t)(R_i^t - b_i^t) \tag{12}$$

where N is number of episodes and T is the length of time budget.

The goal of the agent is to give correct recognition for all the grid cells in the map, and we hope it does not waste time in revisiting the same location, thus we introduce a reward function R_{cls} that seeks to maximize true positive classifications while minimizing false positive classifications at the final time step T, and a reward function $R_{revisit}$ to penalize the re-visitation of the same positions at each time step t:

$$R^t = R^t_{cls} + R^t_{re} \tag{13}$$

$$R^t_{cls} = \begin{cases} 0, & \text{if} \quad t < T; \\ N_{tp}R_{tp} + N_{fp}R_{fp}, & \text{if} \quad t = T \end{cases} \tag{14}$$

$$R^t_{revisit} = \begin{cases} R_{re}, & \text{if} \quad P_t \subseteq \{P_0, P_1, ..., P_{t-1}\} \\ 0, & \text{else} \end{cases} \tag{15}$$

where N_{tp} is the number of true positive classifications, N_{fp} is the number of false positive classifications, R_{tp} and R_{fp} are positive and negative rewards contributed by each of these classifications, respectively, R_{re} is the negative rewards to the re-visitation of a same position during one episode.

VI. Simulation Framework

To train and evaluate our model, we require a framework for performing actions and perceiving their outcomes in a 3D environment. Thus we use AirSim [23], a plugin for Unreal Engine, which captures realistic flight dynamics and can be integrated with customized 3D scenes.

A. Urban Scenes with Humans on Streets

In this work, we focus on human detection in urban streets. We develop 10000 different scenes with realistic building and street layouts that are created through Esri CityEngine [28]. Each scene data is a square of size 50m×50m. We use an occupancy map of size 10×10 to represent where the humans are present in this scene, thus each cell of the occupancy map represents an area of 5m×5m. We use MakeHuman, Blender, and the CMU Motion Capture Dataset to synthesize 3D human models in realistic poses. Then we add those human models to the streets. Figure 1 shows an example of the synthetic urban streets with pedestrians on roads.

B. Learning Setting

We select a subset of the scenes uniformly at random to train and evaluate. We have three train/ test settings:

- Setting 1: train/ test = 20/ 10 scenes, each scene has 2 grid cells occupied by human.
- Setting 2: train/ test = 20/ 10 scenes, each scene has 5 grid cells occupied by human.
- setting 3: train/ test = 20/ 10 scenes, each scene has 10 grid cells occupied by human.

For each setting, the scenes in training and the scenes in testing have no overlap.

The start position of the UAV is set to the center of the scene, 70 meters above the ground plane. The observations are images of resolution 224 × 224 taken by the agent's RGB camera in its first person view. We use a constant step length (10 meters) and with the camera optical axis fixed to be perpendicular to the ground plane.

VII. EXPERIMENTAL SETUP AND RESULTS

The aim of our experimental evaluation is to study the following questions:

- What aspects of the proposed Human Occupancy Estimation task and dataset motivate and benefit active vision?
- Is the proposed end-to-end learning of fusion and policy framework effective and efficient compared to the baselines?

We study the first question by examining the single view performance both for a single cell occupancy recognition and a map occupancy estimation, and conclude that the factors influence performance both for a single cell recognition and multi cells recognition are hard to hand code, which motivates a data-driven method for both the information fusion mechanism and view selection.

We study the second question by extensively evaluating our model against 5 baselines and two methods with an oracle fusion module.

A. Evaluation Metric for Human Occupancy Map Estimation

To evaluate the performance for one map of size $N \times N$, we use F1 score and accuracy. The classification result for one cell is considered correct if $\overline{p}(y_i = y_i^{gt}) > 0.5$, where y_i^{gt} is the ground truth label for grid g_i.

The number of true positive classified cells, true negative classified cells, false positive classified cells and false negative classified cells are denoted as N_{tp}, N_{tn}, N_{fp} and N_{fn}.

$$acc = \frac{N_{tp} + N_{tn}}{N \times N} \tag{16}$$

$$\mathrm{F1} = \frac{2 \times N_{tp}}{2 \times N_{tp} + N_{fp} + N_{fn}} \tag{17}$$

To evaluate performance on a set of maps, we use average accuracy and average F1 score across the maps as the metric:

$$Aacc = \frac{1}{N} \sum_{i=1}^{N} acc_i \tag{18}$$

$$A\mathrm{F1} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{F1}_i \tag{19}$$

where acc_i is the accuracy of map i, and $\mathrm{F1}_i$ is the F1 score of map i.

B. Hyper-parameters and Implementation Details

All the parameters $\Theta = [\theta_f, \theta_c, \theta_w, \theta_a, \theta_e, \theta_{st}]$ are optimized using ADAM [29], θ_f, θ_c are optimized using $\alpha = 0.000001$, learning rate of 0.0001. For optimizing θ_w, we use $\alpha = 0.000001$, learning rate of 0.000001. For optimizing the REINFORCE objective, i.e., to learn $\theta_a, \theta_e, \theta_{st}$, we use $\alpha = 0.000001$, learning rate of 0.00001. The mini-batch size for the first stage training is 32, and 4 for the second stage training.

C. One View Performance on a Single Grid and a Map

Fig. 4 shows the accuracy of grid classification vs. the size of the grid cell in the image frame. Formally, the size of grid cell in the images frame is defined as : $F_{size}(\left\{p_i^{C_n}\right\}^t_{n=1...4})$, where $F_{size}(P)$ counts the number of pixels within the shape composed by the points set P, and $\left\{p_i^{C_n}\right\}^t_{n=1...4}$ are the pixel coordinates of the four corners of grid g_i in the image frame at time step t.

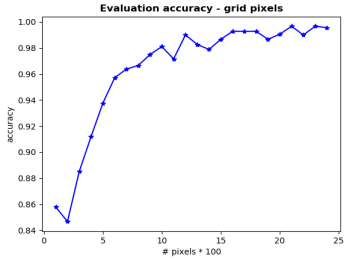
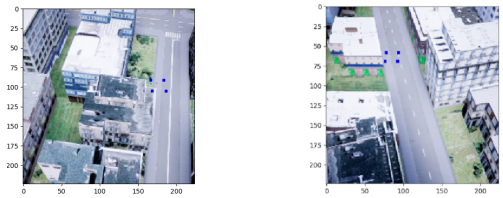


Fig. 4: Accuracy of a single view for grid cells vs. the size of mapped grid cells.

In general, the larger mapped grid cells lead to higher accuracy, while the accuracy decreases when the grid size increases from 100 to 200. This shows that the single grid classification performance is not merely depends on the size

of the grid cell in the observation. Fig. 5 can provide an explanation of why smaller grid cells in the observation frame can lead to correct classification results while larger grid cells can lead to wrong classification results. The marked grid cell in Fig. 5(a) is of 300 pixels while the human in that cell looks smaller than the human in the marked grid cell of 200 pixels in Fig. 5(b).



(a) a grid cell of 300 pixels (b) a grid cell of 200 pixels

Fig. 5: Example of a larger grid cell leads to wrong classification while a smaller grid cell leads to correct classification.

TABLE I: Single view performance vs. Pedestrian density

Number of cells occupied by human	Aacc↑	AF1↑
1-2	0.896	0.201
3-4	0.892	0.356
5-6	0.883	0.445
7-8	0.872	0.494
9-10	0.870	0.562
11-12	0.873	0.614

Table. I shows the accuracy and the F1 score for maps of different human density in terms of a single initial view. The testing set for the one view evaluation on whole map are scenes randomly selected from 10000 scenes, with each range containing 80 scenes.

Thus the benefits that are expected from extra views are different for maps of different human density. We hypothesize that we need to develop different fusion and view selection methods to deal with maps of different human density in future work.

D. Baselines

We first introduce some other multiple measurements fusion methods:

- Bayes rule: this is the update rule adopted by the original occupancy map framework, i.e. [3].
- Average: this fusion method uses the averaged probability across T measurements as the final prediction, i.e., $\overline{p}(y_i = 0) = \frac{1}{T}\sum_{t=1}^{T} p\left(y_i^t = 0\right)$.
- Fusion based on the size of grid cell: this fusion method takes the classification result for grid cell g_i as the one whose number of pixels of the mapped grid cell in the image frame is the largest one among all T measurements, i.e., $\overline{p}(y_i = 0) = p\left(y_i^t = 0\right)$, where $t = \underset{t\in[1,T]}{\operatorname{argmax}} F_{size}(\left\{p_i^{C_n}\right\}_{n=1\ldots4}^{t})$. This fusion method is to verify that the performance of the classifier is not only dependent on the size of the grid cell. Larger grid cells in the image frame do not necessarily guarantee better classification performance. Thus simply taking the classification result whose input is of the largest number of pixels does not lead to the best fusion results.
- Oracle fusion: this fusion method marks the upper performance bound of the fusion module by considering the final classification result for one grid as correct if the grid is classified correctly at least one time during the T measurements, otherwise the final classification result is wrong at time T.

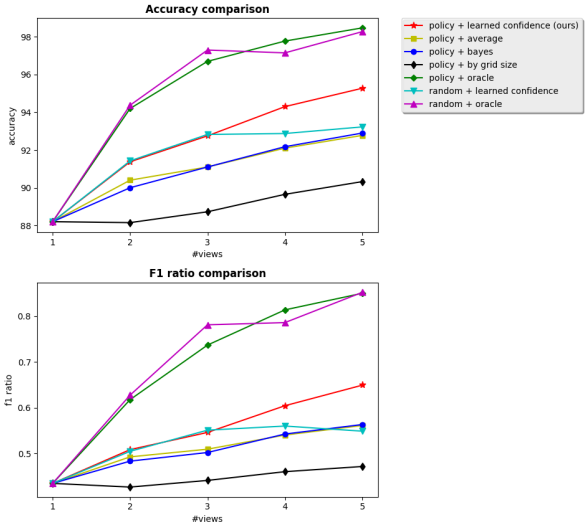


Fig. 6: Evolution of performance over time for our method, vs. baselines.

We compare our proposed end to end learning of the confidence based fusion module and the policy network to the following baselines:

1) Single view (takes the classification result from the initial view);
2) Bayes rule fusion plus policy;
3) Average fusion plus policy;
4) Fusion based on the size of grid cell plus policy;

For baselines (2) - (4), the policy shares the same architecture and interaction mode with the fusion module with ours.

5) Learned confidence based fusion plus random policy. This method uses the same fusion architecture as ours and interaction mode with a random policy which samples an action from a uniform distribution over the action space at each step;
6) Oracle fusion plus policy.
7) Oracle fusion plus random policy.

E. Results

Table. II shows the results of our methods compared with baselines in the three learning settings. Fig. 6 shows the evolution of performance over time for our method and baselines in the learning setting 2 where each scene data have 5 grid cells that are occupied by a person. Our method outperforms baselines (2)-(4) in both metrics demonstrating the effectiveness of the proposed fusion module. Our method

TABLE II: Comparison with other approaches on maps of different human density given $T = 5$

Number of cells occupied by human	2		5		10	
	Aacc(%)↑	AF1↑	Aacc(%)↑	AF1↑	Aacc(%)↑	AF1↑
single view	92.150	0.377	88.200	0.434	89.025	0.606
oracle fusion + random policy	98.825	0.819	98.275	0.852	98.400	0.920
learned confidence fusion + random policy	95.700	0.501	93.225	0.549	92.975	0.691
average + policy	95.350	0.514	92.775	0.561	93.950	**0.746**
bayes rule + policy	94.900	0.471	92.900	0.563	94.000	0.743
by grid size + policy	93.150	0.420	90.325	0.472	90.275	0.632
(Ours) learned confidence fusion + policy	**96.450**	**0.555**	**95.275**	**0.649**	**94.425**	0.737
oracle fusion + policy	99.250	0.862	98.475	0.850	98.475	0.924

outperforms baseline (5) demonstrating the effectiveness of the policy network.

All the methods outperform the single view baseline, validating the necessity of multi-view acquisition for accurate occupancy map estimation.

VIII. CONCLUSION

We presented a novel approach for vision-based active object search in scenes. The core issues in active object search include how to combine information acquired across multiple views and how to plan a sequence of views that will lead to accurate object detection in a timely fashion. Our proposed model addresses these two issues via a combined learning strategy and model. We demonstrated that this model is effective at acquiring and aggregating information. A human detection from UAV imagery simulation testbed was developed in order to validate our methods. We believe that this approach and testbed can serve as fruitful grounds for future research into this important active vision task.

REFERENCES

[1] S. D. Roy, S. Chaudhury, and S. Banerjee, "Active recognition through next view planning: a survey," *Pattern Recognition*, vol. 37, no. 3, pp. 429–446, 2004.

[2] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, no. 6, pp. 46–57, 1989.

[3] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI magazine*, vol. 9, no. 2, p. 61, 1988.

[4] J. D. Adarve, M. Perrollaz, A. Makris, and C. Laugier, "Computing occupancy grids from multiple sensors using linear opinion pools," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4074–4079.

[5] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 945–953.

[6] B. Shi, S. Bai, Z. Zhou, and X. Bai, "Deeppano: Deep panoramic representation for 3-d shape recognition," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2339–2343, 2015.

[7] D. Wilkes and J. K. Tsotsos, "Active object recognition," in *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*. IEEE, 1992, pp. 136–141.

[8] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966–1005, 1988.

[9] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[10] M. Malmir, K. Sikka, D. Forster, J. R. Movellan, and G. Cottrell, "Deep q-learning for active recognition of germs: Baseline performance on a standardized dataset for active learning." in *BMVC*, 2015.

[11] D. Jayaraman and K. Grauman, "Learning image representations tied to ego-motion," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1413–1421.

[12] M. Malmir and G. W. Cottrell, "Belief tree search for active object recognition," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 4276–4283.

[13] E. Johns, S. Leutenegger, and A. J. Davison, "Pairwise decomposition of image sequences for active multi-view recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3813–3822.

[14] D. Jayaraman and K. Grauman, "Look-ahead before you leap: end-to-end active recognition by forecasting the effect of motion," in *European Conference on Computer Vision*. Springer, 2016, pp. 489–505.

[15] W. Luo, P. Sun, F. Zhong, W. Liu, and Y. Wang, "End-to-end active object tracking via reinforcement learning," *arXiv preprint arXiv:1705.10561*, 2017.

[16] P. Ammirato, P. Poirson, E. Park, J. Košecká, and A. C. Berg, "A dataset for developing and benchmarking active vision," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1378–1385.

[17] X. Ye, Z. Lin, H. Li, S. Zheng, and Y. Yang, "Active object perceiver: Recognition-guided policy learning for object searching on mobile robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6857–6863.

[18] D. Jayaraman and K. Grauman, "Learning to look around: Intelligently exploring unseen environments for unknown tasks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1238–1247.

[19] S. K. Ramakrishnan and K. Grauman, "Sidekick policy learning for active visual exploration," in *European Conference on Computer Vision*. Springer, 2018, pp. 424–442.

[20] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3357–3364.

[21] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2616–2625.

[22] L. Paletta and A. Pinz, "Active object recognition by view integration and reinforcement learning," *Robotics and Autonomous Systems*, vol. 31, no. 1-2, pp. 71–86, 2000.

[23] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and service robotics*. Springer, 2018, pp. 621–635.

[24] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[26] S. Daftry, S. Zeng, J. A. Bagnell, and M. Hebert, "Introspective perception: Learning to predict failures in vision systems," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 1743–1750.

[27] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[28] "Esricityengine: http://www.esri.com/software/cityengine."

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.