Exploiting Natural Language for Efficient Risk-Aware Multi-Robot SaR Planning

Vikram Shree, Beatriz Asfora, Rachel Zheng, Samantha Hong, Jacopo Banfi, and Mark Campbell

***Abstract*—The ability to develop a high-level understanding of a scene, such as perceiving danger levels, can prove valuable in planning multi-robot search and rescue (SaR) missions. In this work, we propose to uniquely leverage natural language descriptions from the mission commander in chief and image data captured by robots to estimate scene danger. Given a description and an image, a state-of-the-art deep neural network is used to assess a corresponding similarity score, which is then converted into a probabilistic distribution of danger levels. Because commonly used visio-linguistic datasets do not represent SaR missions well, we collect a large-scale image-description dataset from synthetic images taken from realistic disaster scenes and use it to train our machine learning model. A risk-aware variant of the Multi-robot Efficient Search Path Planning (MESPP) problem is then formulated to use the danger estimates in order to account for high-risk locations in the environment when planning the searchers' paths. The problem is solved via a distributed approach based on Mixed-Integer Linear Programming. Our experiments demonstrate that our framework allows to plan safer yet highly successful search missions, abiding to the two most important aspects of SaR missions: to ensure both searchers' and victim safety.**

***Index Terms*—Multi-modal perception for HRI, multi-robot systems, search and rescue robots.**

I. Introduction

ACCURATE scene awareness is the keystone for success in search and rescue (SaR) missions. The deployment of robots in the World Trade Center disaster pinpointed limitations in different aspects of robotic systems, including human robot collaboration [1]. For the past two decades, sensors in particular have evolved in number and variety, and are now able to generate gigabytes of data within seconds, and even extract important features autonomously. However, rigorous analysis and summarizing of the large amount of data about a scene in order to infer high-level understanding of the surrounding world is still a work in progress. Errors propagate across and down the multirobot system pipeline, with detrimental impact to the SaR mission performance.

Manuscript received October 16, 2020; accepted February 9, 2021. Date of publication March 1, 2021; date of current version March 19, 2021. This letter was recommended for publication by Associate Editor M. Ani Hsieh and Editor J. Fu upon evaluation of the reviewers' comments. This work was supported by the National Robotics Initiative (NRI) program of the National Science Foundation, Award #1830497. *(Vikram Shree and Beatriz Asfora contributed equally to this work.) (Corresponding author: Vikram Shree.)*

The authors are with the Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 208016 USA (e-mail: vs476@cornell.edu; ba386@cornell.edu; rz246@cornell.edu; sh974@cornell.edu; jaco.banfi@gmail.com; mc288@cornell.edu).

Digital Object Identifier 10.1109/LRA.2021.3062798

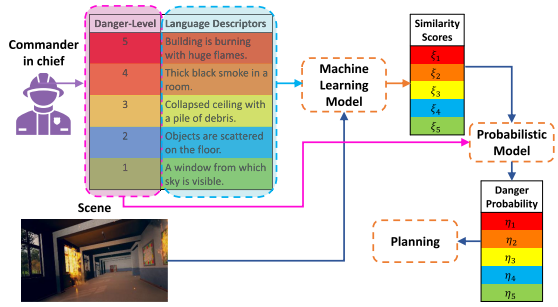

Fig. 1. Scene danger perception and planning pipeline. A set of descriptors and corresponding danger levels, provided by the mission commander, is used for estimating danger probability distribution, on a 5-point scale. This information is used by the planning module to plan safe paths for the agents.

Prior work in robotic perception has focused on inferring low-level information about the environment, for example, building occupancy grid maps [2] or mapping unique landmarks [3]. These low-level attributes are indeed relevant to build a representation of the world that is suitable for navigation. However, planning for a team of agents typically requires humans to make decisions based on high-level attributes of the environment. These include the notion of "danger," for which the ability to map low-level aspects of the scene into a high-level and succinct representation is still an open question in the robotics community. In this work, we address this problem by proposing a systematic approach for inferring scene danger from visio-linguistic inputs to enable high-level planning for a team of agents.

Our approach, sketched in Fig. 1, asks users for descriptions that they feel are important to characterize scene danger. Descriptions are matched against images seen from the robot's camera by leveraging a machine learning model, and the obtained similarity scores are used to keep up-to-date a probability distribution describing danger in different areas of the environment. This information, in turn, is used to plan more informed paths for the searchers. In summary, this letter makes the following novel contributions:

1) The adaptation of a state-of-the-art deep neural network [4] to assess similarity between the descriptions and images in the scene. To facilitate the use of the network in SaR missions, we introduce a novel dataset, consisting of language descriptions for synthetic disaster scenes taken from the DISC dataset [5].
2) An intuitive probabilistic model for estimating scene danger by fusing similarity scores obtained from various user descriptions and multiple images at a scene. Compared

against a system (e.g. a classifier) that is directly trained on a priori notions of "danger," our approach can adapt to the needs of different missions (e.g. fire, earthquake, or radioactive incident) without having to change any of its parameters.

3) The introduction of a risk-aware version of the Multi-robot Efficient Search Path Planning (MESPP) problem [6] which can leverage scene understanding to ensure agents' safety by accounting for each heterogeneous agent's danger tolerance. This is an online problem, which we solve via a distributed planning approach based on variants of state-of-the-art Mixed-Integer Linear Programming (MILP) models [7].

Extensive numerical and realistic ROS/Gazebo simulations show that our holistic approach to multi-robot SaR planning enforces the safety of heterogeneous agents under different notions of risks specified by the user, while maintaining performance in terms of time required to locate the victim.

II. Related Work

A. Collaborative Search and Rescue Missions

The challenges encountered in SaR missions depend upon the operational environment, which can be divided into three main categories: maritime, urban, and wilderness [8]. In this letter, we focus on SaR in urban environments. Robots can go into more dangerous areas to avoid risking human lives, however they still need to leverage the human expertise for scene understanding. Yazdani *et al.* [9] studied how high-level instructions from a human can be used to plan actions for the robot during SaR missions. In order to bridge the gap between scene perception and decision-making, researchers have proposed ontology models [10] that consist of rules to represent the environment and action space. However, these approaches have been only shown to work in simple settings and are prone to failure in more complex, uncertain scenarios, which are often encountered in SaR missions.

B. Language-Based Scene Assessment

In a typical rescue mission, human rescuers talk with each other on a low-bandwidth radio channel since it has virtually unlimited range. The human brain is great at interrelating data from visual and language domains, but when it comes to neural networks, this task becomes more challenging due to the lack of a one-to-one relationship between the two inputs.

Modern deep neural architectures [4], [11] tend to address this challenge by extracting high-level features from each one of the inputs before jointly reasoning about them. The benefits of these models, however, have only been tested on datasets describing serene environments (e.g. [12]) because of the constraints associated with replicating a realistic SaR scenario. In this work, we leverage photo-realistic synthetic images from disaster scenes [5] to conduct a large-scale survey. This allows us to obtain language data, which is vital for training state-of-the-art visual-language models before applying them in SaR missions.

Compared to other approaches for processing visio-lingual data that can be found in the computer vision community, like visual question answering [13] and visual commonsense reasoning [14], caption-based image retrieval [15] is the most relevant from the standpoint of this work. This refers to the task of identifying an image from a large pool given a caption describing its content, thus, requiring to estimate similarity between images and text data. There is a rich line of work towards mapping images and sentences to a common semantic space for assessing image-text similarity. In this letter, we compare the usage of SCAN [11] and ViLBERT [4] on a new dataset built on top of the disaster scenes contained in the DISC dataset [5], due to their superior performance reported in the literature [4].

C. Multirobot Search

We assess the advantages brought by our danger estimation pipeline by formulating and evaluating a risk-aware version of a famous robotic search problem, the Multi-robot Efficient Search Path Planning (MESPP) problem introduced by Hollinger *et al.* [6]. In the original version of the MESPP problem, a team of robots is deployed in a graph-represented environment with the aim of locating a moving non-adversarial target within a given deadline. In this new online version, graph vertices are associated with probabilistic danger estimates, and robots are heterogeneous in terms of their tolerance to hazardous conditions. This is different from the settings that can be found in the literature, which either focus on static threats [16], or consider dynamic threats but in presence of a single agent that has to reach a given goal location while maximizing its chances of survival (without optimizing a second performance metric) [17].

III. Proposed Architecture

Let us define five intuitive danger levels : low, moderate, high, very high, and extreme, and assign a value $l \in \mathcal{L}$ to each, respectively $\mathcal{L} = \{1, \ldots, 5\}$. We assume that an experienced user (the commander in chief) is able to provide at least one sentence characterizing a description of each danger level (e.g., "a room is filled with fire," associated with $l = 5$). Our pipeline consists of three layers, as shown in Fig. 2. Layer I is tasked with estimating similarity between the user's descriptions and images collected in a scene, providing a matching score for each image-description pair. The scores of all images are combined into a probabilistic estimate of the danger level (Layer II), which is in turn used for planning an efficient risk-aware search for the team (Layer III). This plan is then sent to the team of robots operating in the environment, where they search for victim and acquire new images. Each layer is described in detail in the following sections.

IV. Text-Image Similarity Assessment

As a first step to establish the user-perceived danger level of the scene, we start with determining similarity between the provided language descriptors and the scene. In this work, a scene consists of a set of images acquired online by the robot. The robot can compute a *similarity score* ξ for each (image,

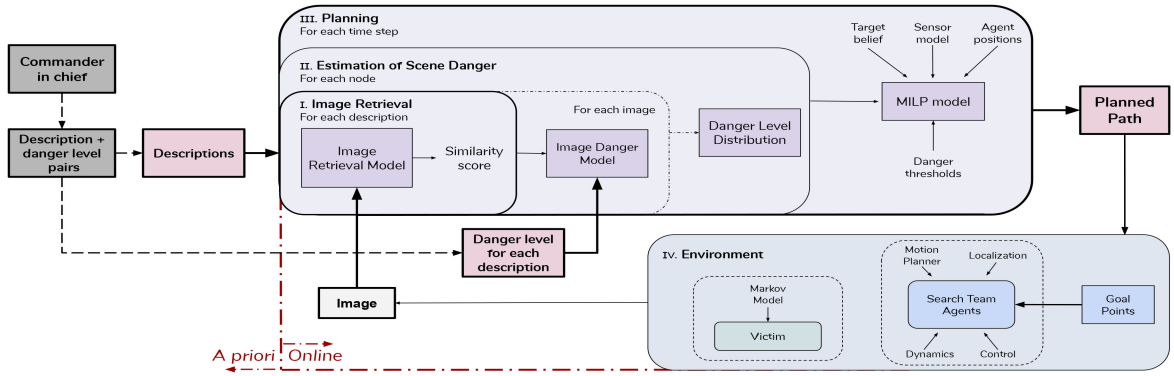

Fig. 2. System layers: text-image similarity assessment (I), scene danger estimation (II) and multi-robot planning (III).

descriptor) pair. This can be done with one of the following deep learning architectures.

1. Scan

Stacked Cross Attention for Image-Text Matching (SCAN) [11] uses a two-stage attention architecture for calculating text-image similarity. First, the word and image are converted into a set of word and image features, respectively. This is followed by calculating a cosine similarity matrix for each possible word-image feature pair. The first attention stage attends to image regions w.r.t each word in the sentence. The next stage compares the words based on the attended image vector and decides the importance of each word. The final pooling layer outputs a similarity score.

2. Vilbert

Visual-Language BERT (ViLBERT) [4] introduces two separate streams for processing vision and language data that can communicate with each other via co-attention transformer layers. After having converted word and image into features, ViLBERT processes the features with transformer blocks independently before they could interact with each other. The co-attention layers produce attention features for each modality conditioned on the other. The final pooling layer outputs a similarity score.

A. A Novel Emergency Scene Dataset: DISC-L

The DISC-Language dataset[1] is built on top of DISC dataset [5], consisting of 300 K photo-realistic synthetic images taken in several environments (like office and subway) with two types of damage conditions: collapse and fire. The images are captured from a stereo-camera, moving along a pre-defined path in a 3D model world. To reduce redundancy in images, we uniformly sample 1 out of every 150 images from the left camera, yielding a set of 1000 images.

We conduct an online survey on Amazon Mechanical Turk (AMT) to collect sentence descriptions for each image. AMT is a global service enabling us to reach a huge participant pool which favorably introduces diversity in the collected data. Each task requires the user to read the instructions, look at annotated examples, and describe important aspects of a given image with two sentences in English. To incorporate diversity in language, each image was labelled by four unique workers, yielding a total of 4000 descriptions. Furthermore, to ensure high quality language input from the users, the responses are first filtered automatically and then manually approved to remove unsatisfactory descriptions. A few examples are shown in Fig. 3. More examples of accepted and rejected responses can be found on the DISC-L Github page.

- The view of a corridor where on the right you see windows with closed blinds.
- In the middle of the corridor all the furniture is lying on the floor and overturned.

- A number of houses could be seen along the roadside. The place looks green.
- The houses are lit on fire and looks like a big accident has occurred.

Fig. 3. A few sample descriptions for images in the DISC-L dataset, collected through AMT survey.

In summary, DISC-L dataset consists of rich and diverse descriptions of emergency situations from 720 unique Amazon users. There are a total of 3386 unique words in our proposed dataset. The word-count ranges from 10 words to 70 words per description, with a median of 20. A key observation is that the most frequently used words (fire, smoke, flame, dark etc.) are related to situations commonly encountered during SaR mission.

B. Caption-Based Image Retrieval Performance

1) Dataset: The DISC-L dataset is used to evaluate the performance of SCAN and ViLBERT. The dataset is divided into train, validation and test sets such that all images corresponding to a environment remain in the same set. The train set consists of 828 images from six environments; validation has 75 images from three environments; and test has 97 images from two environments.

2) Training: SCAN is pre-trained for caption-based image retrieval on the Flickr30 K dataset [12]. A bottom-up attention

[1]Freely available for academic purposes at https://github.com/vikshree/DISC-L.git

TABLE I
SCAN VS VILBERT MODELS EVALUATED ON DISC-L

Model	Training type	R-1	R-5	R-10
SCAN	Pre-trained	7.5	24.0	39.2
	Fine-tuned on DISC-L	7.7	30.4	48.2
ViLBERT	Pre-trained	9.7	39.3	59.7
	Fine-tuned on DISC-L	**19.4**	**52.0**	**70.4**

network [18] is used to compute a 36×2048 dimensional feature vector for each image, before feeding it to the SCAN network. Finally, the data batch is shuffled online to get negative sentence-image pairs, and a triplet loss function is used to fine-tune the network on the DISC-L dataset. Triplet loss is a ranking-based loss function, commonly used in image-text matching task where distance between an anchor is minimized from the truth value (see [11] for details).

ViLBERT is pre-trained for multiple visual-language tasks on 12 different datasets [19]. A combination of pre-trained Faster R-CNN and ResNet [20] is used to compute a 101×2048 dimensional feature vector for each image, which is fed to the ViLBERT network. Following the authors' approach [4], we adopt a 4-way multiple choice training process where for each sentence-image pair in the dataset, three distractor pairs are sampled with no correspondences. Finally, we fine-tune the model on DISC-L dataset with a cross-entropy loss.

3) Results: We use standard rank-based performance metrics (R-1, R-5, R-10) to evaluate the models, where R-k denotes the proportion of times the correct image is present in the top k likely hypotheses for a description.

Table I shows the results. Fine-tuning the models on DISC-L significantly improves the retrieval performance for both SCAN and ViLBERT, since the pre-trained models have never encountered danger-related images or sentences. Also, we observe that ViLBERT outperforms SCAN by a margin of 150%, 100% and 79% for R-1, R5 and R-10 respectively. The superior performance of ViLBERT can be attributed to its more complex model architecture and the use of higher dimensional feature representation for the images. Therefore, we use the ViLBERT model in the remainder of the letter.

V. SCENE DANGER ESTIMATION

We formulate the task of estimating a probability mass function over the danger space $\mathcal{L} = \{1, \ldots, 5\}$, given a set of language descriptors from a user and a scene as a probabilistic inference task. A scene consists of a set of r local images taken from a given region of the environment: $I = \{i_1, i_2, \ldots, i_r\}$.

For notational simplicity, let us assume that a scene consists of a single image. A single set of descriptions is not sufficient to assess the danger level in different hazard conditions. Hence, our model allows the commander in chief to specify multiple descriptions for each level, which can then be grouped together based on m danger "types". For example, those related to descriptions of a fire. In symbols, $S = \{S_1, S_2, \ldots, S_m\}$, where each set $S_j = \{s_j^1, \ldots, s_j^5\}$ contains a language descriptor for each of the 5 danger levels related to the j-th danger type. Given a single image i and the full set of language descriptors S, the

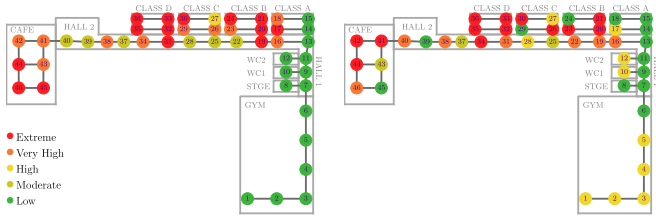

Fig. 4. Graph abstraction of the school environment. Each vertex $v = 1, \ldots, 46$ is color-coded based on its danger level in NFF scenario. Left: Human reasoned ground truth; Right: Estimate using 5% of images and fire descriptors.

ViLBERT model is used to generate a matrix of similarity scores, denoted by $\Xi = \{\{\xi_1^1, \ldots, \xi_1^5\}, \ldots, \{\xi_m^1, \ldots, \xi_m^5\}\}$. To reduce the impact of noise, the scores are converted into one-hot vectors $\mathbf{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_m\}$ based on a threshold θ, such that:

$$\mathbf{y}_u^l = 1 \text{ if } \xi_u^l \geq \theta \text{ else } 0 \; \forall \, l \in \mathcal{L}, \; u \in \{1, \ldots, m\}. \quad (1)$$

The posterior distribution of danger level for the scene $P(D|\Xi, S, I)$, denoted by the vector $\boldsymbol{\eta} = [\eta_1 \; \eta_2 \cdots \eta_5]$, is calculated based on the frequency of samples in the data as:

$$\eta_l = \frac{1}{\rho} \sum_{u=1}^{m} \mathbf{y}_u^l \; \forall l \in \mathcal{L}, \quad (2)$$

where $\rho = \sum_{u=0}^{m} \sum_{l \in L} \mathbf{y}_u^l$ is the normalizing constant. For the more general case where a scene consists of r images, we simply include one-hot vectors obtained from each one of the images while calculating η_l in Eq. (2).

To evaluate our danger estimation approach, we use the School environment from the DISC dataset [5] because of its relatively larger size and create the graph structure shown in Fig. 4, containing $n = 46$ vertices. Each vertex in the graph is associated with a set of local images collected from its immediate surroundings. The DISC dataset allows each vertex to be defined with images containing three different types of danger: none (N), collapse (C), and fire (F). Three 'environments' are designed for simulation based on a predefined proportion of hazards: 'NCF' denotes environments where vertices are associated with hazardous images in equal proportion; 'NFF' denotes environments with no C-type danger and 2/3 of vertices with F-type danger, etc. We introduce a set of descriptions for assessing collapse and fire danger (see the DISC-L dataset Github page for details). For example, the description "The room is engulfed in huge flames" describes a fire danger level $l = 5$. Ground truth danger values l_v for each vertex v are obtained by having a human user reason about danger based on the descriptions and the images associated with vertex v.

We use the average Bhattacharya coefficient (BC)[21], a standard metric to quantify the disparity between two discrete probability distributions, to measure the closeness of the estimates with the ground truth, computed as:

$$BC = \frac{1}{n} \sum_{v=1}^{n} \sqrt{\eta_{l_v, v}}, \quad (3)$$

where $\eta_{l_v,v}$ is the probability corresponding to the ground-truth danger level l_v at vertex v. The results are shown in Table II.

TABLE II
COMPARISON OF AVERAGE BC FOR ESTIMATED DANGER DISTRIBUTION WITH DIFFERENT DESCRIPTIONS, FOR THREE SCENE TYPES

Scene Type	F-Descriptors	C-Descriptors	FC-Descriptors
NFF	**0.63**	0.42	0.59
NCC	0.30	**0.49**	0.47
NCF	0.47	0.49	**0.53**

First, we observe that the danger estimates for all environments are most accurate when the descriptions corresponding to that specific environment are used. For example, if it is a fire hazard (NFF), using only fire-related descriptions yields the best danger estimates ($BC = 0.63$). If there are both fire and collapse vertices (NCF), then using a combination of both descriptors yields the best danger estimates ($BC = 0.53$). Furthermore, the average BC obtained from the '*uniform-prior*' baseline is 0.45, which is lower than the best estimates for each one of the scene-types: NFF (0.63), NCC (0.49), and NCF (0.53). Thus, we conclude that customizing descriptions, depending on the hazard, is indeed beneficial for danger estimation. It should be noted that one can leverage homogeneous domain adaptation techniques [22] to minimize any performance deterioration when transferring the system to the real world.

VI. MULTI-ROBOT PLANNING

A. Risk-Aware MESPP Problem

We now introduce a risk-aware version of the Multi-Robot Search Path Planning (MESPP) problem [6]. In the original MESPP problem, a team of robotic agents $A = \{1, \ldots, m\}$ is deployed in a graph-represented environment $G = (V, E)$, with the aim of locating a possibly moving non-adversarial "target" (e.g. a victim) within a given deadline τ. Time $T = \{1, .., \tau\}$ evolves in discrete steps, and both the agents and the target can either stay at the same vertex or reach a neighbor vertex between two subsequent steps, for vertices $V = \{1, \ldots, n\}$. Agents can communicate with each other at all time-steps.

The target's probable motion in the graph is encoded by a Markovian matrix $\mathbf{M} \in [0, 1]^{n \times n}$. At each step t, the target's state, resulting from its interactions with agents executing a set of joint paths $\boldsymbol{\pi} \in \mathcal{P}$, is represented by the belief vector

$$\mathbf{b}^{\boldsymbol{\pi}}(t) = [b_{\mathrm{c}}(t),\, b_1(t), \ldots, b_n(t)], \tag{4}$$

where $b_{\mathrm{c}}(t) + \sum_{i=1}^{n} b_v(t) = 1$. The first element, $b_{\mathrm{c}}(t)$, represents the probability that the agents have located the target by time t when following paths $\boldsymbol{\pi}$. The remaining elements $b_1(t), \ldots, b_n(t)$ represent the probability that the target is located in the corresponding vertices at time t.

Detection events are described by matrices $\mathbf{C}^{a,u} \in [0, 1]^{(n+1)\times(n+1)}$, $\forall a \in A, u \in V$. Their effect is to connect the probability of the target being at a particular location with its detection state. In other words, the matrix $\mathbf{C}^{a,u}$ encodes which vertices of the graph fall within the sensing range of agent a, when such is located in vertex u. A belief update equation links the current belief, the target's motion, and the agents' paths $\boldsymbol{\pi}$ with associated detection events as follows:

$$\mathbf{b}^{\boldsymbol{\pi}}(t+1) = \mathbf{b}^{\boldsymbol{\pi}}(t) \begin{bmatrix} 1 & \mathbf{0}_{1\times n} \\ \mathbf{0}_{n\times 1} & \mathbf{M} \end{bmatrix} \prod_{a \in A} \mathbf{C}^{a,\pi^{a,t+1}}. \tag{5}$$

In the original MESPP problem, the goal is to find the optimal path $\boldsymbol{\pi}^*$ that maximizes $\sum_{t=0}^{\tau} \gamma^t b_{\mathrm{c}}(t)$, where $\gamma \in (0, 1]$ is a discount factor.

Our risk-aware version of the MESPP problem is an *online* problem defined as follows. Let us associate a ground truth danger level $l_v \in \mathcal{L}$ for each $v \in V$, and, accordingly, a probability of agent loss for each danger level, $p(\mathrm{loss}|l_v)$. Define for each agent $a \in A$, a fixed nominal danger threshold $\kappa^a \in \mathcal{L}$, which is the expected danger level that agent a is apt to endure throughout the mission. We assume the agents are equipped with a danger estimation module they can leverage for estimating danger level distributions $\boldsymbol{\eta}_v^t$, for each vertex v and step t. Such a module provides estimates at $t = 0$ simply based on a fixed prior, uniform by default. Once an agent visits a vertex v for the first time, it is allowed to update the distribution of v (for example, by leveraging the method of Section V) and, possibly, that of neighboring vertices. We introduce two risk-aware variants of MESPP, based on different additional path constraints, contingent upon available danger information.

Point estimate constraints: at each step t, the current plan of each agent a can not prescribe a visit to a vertex v where the most probable danger level of the vector $\boldsymbol{\eta}_v^t$, denoted by $z_v^t \in \arg\max_{l \in \mathcal{L}} \boldsymbol{\eta}_v^t$ belongs to a danger level strictly larger than κ^a.

Cumulative probability constraints: define an agent's required danger confidence, $\alpha^a \in (0, 1]$ as the cumulative probability of estimated danger up to that agent's threshold κ^a. At each step t, the current plan of each agent a can only include vertices where the cumulative danger distribution, denoted by $H_v^{a,t} = \sum_{l=1}^{\kappa^a} \eta_{l,v}^t$ is equal or higher than α^a. This allows to express more nuanced constraints since we need to consider all danger probabilities up to an agent's danger threshold κ^a.

In both cases, the problem objective remains locating the victim as soon as possible, considering that agents might be lost along the mission according to $p(\mathrm{loss}|l_v)$. Note that while we define our thresholds as static parameters, they can also be time-dependent, allowing for a dynamic behavior throughout the mission.

B. MILP Models

Our solution is based on a receding-horizon distributed planning approach, where planning is performed by iteratively solving an extension of the MILP models presented in [7]. We refer the reader to [7] for the complete set of MILP variables and constraints, as well as implementation details, and focus here solely on modeling danger-related constraints.

Analogously to the original MILP models, the binary variable $x_v^{a,t}$ indicates agent a is at vertex v at time t in the computed path. Plans compliant with the point estimated constraints are

obtained by enforcing[2]

$$x_v^{a,t} z_v^t \leq \kappa^a, \forall v \in V, t \in T, a \in A. \tag{6}$$

When dealing with the cumulative probability constraints, we instead impose:

$$H_v^{a,t} \geq x_v^{a,t} \alpha^a, \forall v \in V, t \in T, a \in A. \tag{7}$$

VII. Simulations

A. Environment

We use the School scenario of the DISC dataset [5] with NFF image distribution (see Sec. IV) to validate and analyze the performance of our proposed system in its entirety. Since the available dataset measurements represent what the robot would have collected while moving through the simulated environment, we use the left camera poses to infer our layout, which is then abstracted into the graph shown in Fig. 4. We also use it to re-build the environment in Gazebo [23], an open-source 3D robotics simulator, for more realistic experiments accounting for robot dynamics, asynchronous agents and navigation challenges.

B. Danger Estimation

The general setup is the same for both numerical and qualitative simulations. The human-reasoned ground truth for each vertex (Fig. 4, left), defines the probability of losing the agent $p(\text{loss}|l_v) = 7.47e - 8(l_v - 1)\text{e}^{l_v}\, \forall l_v \in \mathcal{L}$, which yields values between 0.009% and 49.5%.

A partial collection of images (5%) is used for danger distribution estimation. These correspond to the first set of images an agent would see when entering the vertex area, according to the DISC dataset camera pose. Note from Fig. 4 that the estimated danger distribution, and thus the maximum likelihood level (right), do not always match the human-reasoned ground truth (left). Processing all images in a scene requires minutes even with a powerful GPU [24], thus using the first few acquired images for estimation aligns well with practical time limitations.

We divide the school space into neighborhoods, based upon common structures such as walls and doorways (see Fig. 4). We start with an a priori danger distribution for each vertex and the estimated danger distribution for that vertex is made available when an agent visits it for the first time, mimicking online information gathering. This information is then spread to vertices in the neighborhood, if they have not been visited yet. For instance, once an agent reaches $v = 3$, its danger information $(\boldsymbol{\eta}_3^t, z_3^t, H_3^{a,t})$ is passed on to the other vertices in the *gym* neighborhood, $v = 4, 5, 6$, but not $v = 1, 2$ since they have already been visited.

In our simulations, a discrete time-step comprises the following actions: call the planner; get a new plan for the search team; move the agents towards their next goals; retrieve danger data; update danger estimation on visited and neighboring vertices; compute and apply probability of loss given ground truth; update team status (if agents are lost); scan for the victim; and finally, output target detection status.

[2]We remark that a similar effect of usage could be obtained by removing unsuitable states in the definition of legal searchers' paths, prior to planning (see again [7] for details).

C. Planner Parameters

In order to define a challenging initial belief vector, we pick nine random vertices across neighborhoods and assign a uniform probability of victim location among the chosen vertices, assuming such is static when updating the belief. Our team of three agents starts from $v = 1$ (gym entrance) with probability of capture equal to zero, i.e. the victim is not reachable at $t = 0$. For simplicity, we consider a perfect sensing model with detection when both agent and target are in the same location. However, assuming different victim motions and sensing models is straightforward [7].

We use GUROBI [25] on 8 threads of a machine equipped with Intel-Core i9-9900 K and 32 GB RAM to solve our path planning problem. We implement a distributed approach, with a mission deadline and planning horizon of 100 and 14 time-steps, respectively. The solving time is consistently under 0.1 secs for the settings studied in this letter.

D. Configurations

We perform ten sets of experiments, with 1000 instances each. We vary:

1) Planner: without danger constraints (NC), with point estimate (PT) and cumulative probability constraints (PB);

*2) A **P**riori Danger Knowledge:* available to our agents at $t = 0$, either perfect (ground truth) **k**nowledge (PK) or no knowledge at all, where we assume an **u**niform distribution for all vertices (PU);

3) Team Makeup: different threshold combinations; (345) for $\boldsymbol{\kappa} = [3, 4, 5]$, $\boldsymbol{\alpha} = [0.6, 0.4, 0.4]$; (335) $\boldsymbol{\kappa} = [3, 3, 5]$, $\boldsymbol{\alpha} = [0.6, 0.6, 0.4]$; and (333) $\boldsymbol{\kappa} = [3, 3, 3]$, $\boldsymbol{\alpha} = [0.6, 0.6, 0.6]$;

4) Best Case Baseline: danger-free environment (ND), i.e., without probability of loss.

The configurations are denoted in this order: {planner – a priori knowledge – team makeup}. Thus, PB-PK-345 denotes cumulative probability planner, perfect a priori knowledge and team threshold makeup of $\boldsymbol{\kappa} = [3, 4, 5]$, $\boldsymbol{\alpha} = [0.6, 0.4, 0.4]$.

E. Metrics

1) Mission Outcomes: success, target is found within the deadline; abort, all agents are lost; and cutoff, deadline is reached before target is found.

2) Average Mission Time: discrete time step when mission ends, either due to target detection, mission abortion or cutoff.

3) Losses: percentage of missions where agents are lost due to the dangerous environment. To evaluate the safety potential of our framework, we denote as 'Most Valuable Agent' (MVA) the agent we want to protect the most, setting its danger threshold as the lowest among the team. Thus, for the team configuration (345), the MVA is agent $a = 1$ with $\kappa^{\text{MVA}} = 3$; for (335), there are two MVAs, $a = 1, 2$; configuration (333) represents a homogeneous team. For a fair comparison, we consider a MVA loss when *at least* one MVA is lost (analogously for N-MVA).

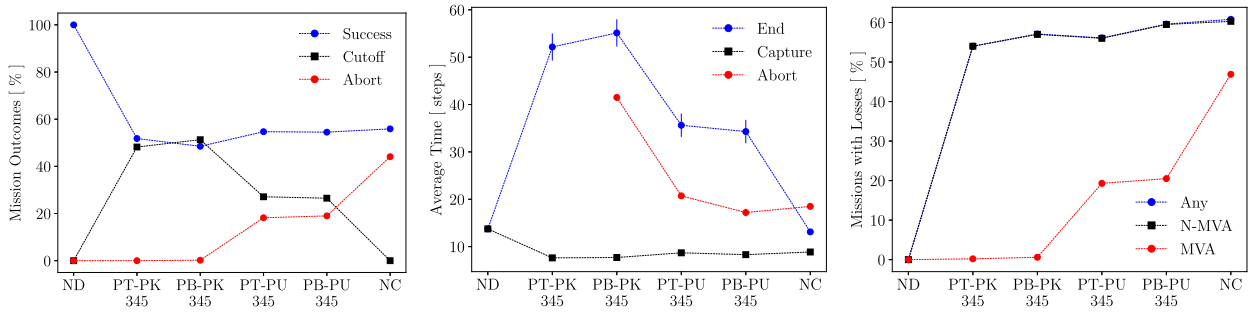

Fig. 5. Numerical simulations with varying *planner* and *a priori danger knowledge*. Planner: point estimate (PT) and cumulative probability (PB) constraints, with $\boldsymbol{\kappa} = [3, 4, 5]$ and $\boldsymbol{\alpha} = [0.6, 0.4, 0.4]$. A priori knowledge: uniform (PU) and perfect (PK), i.e., equal to the human-reasoned ground truth. Additionally, we simulate best (danger-free environment, ND) and worst case (no constraints, NC) scenarios. Left: Mission outcomes. Middle: Mission times. Right: Losses.

F. Numerical Results

The numerical simulation results with varying planner and a priori knowledge are shown in Fig. 5, for an heterogeneous team with danger thresholds $\boldsymbol{\kappa} = [3, 4, 5]$ and $\boldsymbol{\alpha} = [0.6, 0.4, 0.4]$.

The best-case scenario (ND) establishes a baseline of 100% success in the environment studied, which starts to decrease when probabilities of loss and danger constraints are added. The former has the effect of incapacitating the team so they are unable to proceed with the mission, while the later may slow down the search if some plans are considered unsuitable given the agents thresholds.

With perfect knowledge (PT-PK, PB-PK), the team is able to plan accordingly (Fig. 5, left-red): there are only two missions aborted, and our MVA is rarely lost (Fig. 5, right-red). However, as danger estimation becomes less accurate (PT-PU, PB-PU), the planner cannot be so protective, thus the MVA losses increase (Fig. 5, right-red) as well as aborted missions (Fig. 5, left-red), leading to the worst-case scenario, where danger is present but there are no constraints (NC). Note the cause of failure with perfect knowledge (PK) is due to the deadline being reached, while without danger constraints (NC) the mission fails due to abort, which is corroborated by its much shorter average mission time[3] (Fig. 5, middle-blue). The target detection time is roughly the same across configurations (Fig. 5, middle-black), indicating efficiency is still the main goal. Whenever this goal is not achieved, however, the danger constraints still allow the agents to be safer. Meanwhile, if danger constraints are not used and the mission fails, both agents and target are lost.

Results for different team threshold makeups are shown in Fig. 6. The cumulative probabilistic constraints (PB) have a more stable performance than the point estimate (PT) with different teams, as seen in Fig. 6 (left). For a homogeneous team (PB-PU-333), the mission success rate is only slightly lower than for a heterogeneous one (PB-PU-345). This is likely due to the more nuanced thresholding, e.g. 60% confidence that a vertex level is between 1-3, rather than considering a single estimate. However, point estimate is simpler to tune and therefore may perform better in a multimodal danger distribution. For both planners, the main cause of failure is cutoff, particularly for PT-PU-333, which presents a significant longer mission time (Fig. 6, right),

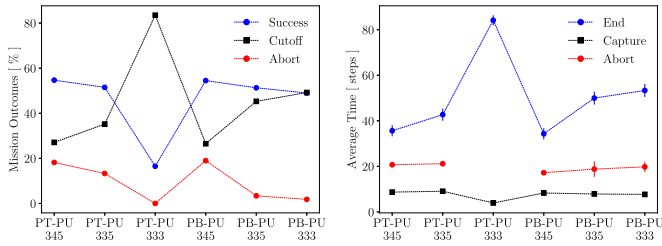

Fig. 6. Numerical simulations with varying team threshold makeups. Left: Mission outcomes. Right: Mission times.

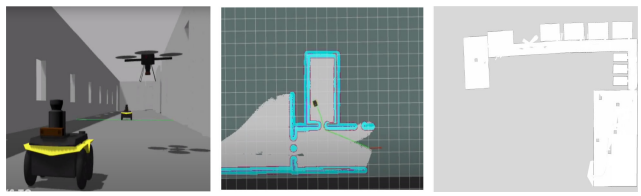

Fig. 7. Environment built in ROS/Gazebo based on the poses extracted from DISC dataset. Left: Structure. Middle: Mapping process. Right: Resultant map.

and lower success rate. On the other hand, the mission abort rate decreases as the team becomes more homogeneous, illustrating the trade-off between protecting the agents and exploring the environment.

G. Qualitative Simulations

We use Gazebo 7 [23] with ROS-Kinetic [26] to incorporate one Hector quadrotor (agent 1) and two Jackal ground robots;[4] (agents 2, 3), as shown in Fig. 7 (left). The simulator's main capabilities of mapping, localization, and autonomous navigation are built off the ROS Navigation Stack. Gazebo simulates the actual robot dynamics, which adds to the possibility of mission failure. The robots can now become inactive either due to danger, or incidents while navigating between vertices, for example getting lost, crashing, or tipping over. If a robot cannot reach its goal vertex before a given time has elapsed, that robot is considered to be inactive, and can no longer participate in the search.

We perform 5 instances per configuration, with each discrete time step corresponding in average to 11.72 secs. The team

[3]Bars denote 95% confidence interval; if not shown, values are ≤ 1 times-step and overlap with marker.

[4]ROS packages: https://github.com/tu-darmstadt-ros-pkg/hector_quadrotor https://clearpathrobotics.com/jackal-small-unmanned-ground-vehicle

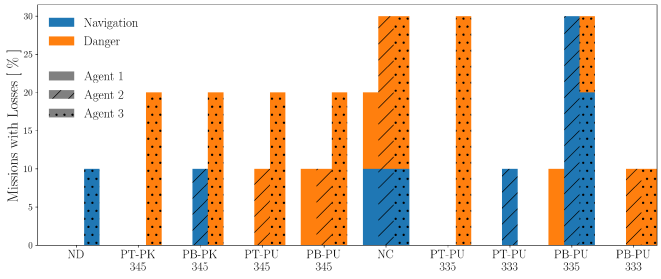

Fig. 8. ROS/Gazebo simulations: percentage of missions where each agent becomes inactive, due to dangerous environment or navigation challenges.

performance follows the trend of the numerical simulations, with similar capture time across configurations. The best-case scenario (ND) presents 100% success, followed by 80% for heterogeneous teams (PT/PB-PU-345), 60% for perfect a priori knowledge (PT/PB-PK-345), and 40% for the others (PT-PU-335, PB-PU-335/333); particularly for PT-PU-333, none of the missions are successful due to cutoff. There are less agent losses and no abort missions with PK, but also less exploring, which results in a lower success rate than PU in these particular instances. All mission failures when employing constraints are due to cutoff, except for one abort instance in PB-PU-345/335.

Fig. 8 shows the percentage of missions where each agent becomes inactive, partitioned into losses caused by Danger and Navigation. There is an expected increase in agent loss when accounting for navigation, particularly for agents 2 and 3 – which can be attributed to the fact that these are ground robots, and thus more likely to run into each other or get stuck. Our results suggest that a risk-aware planner can reduce losses on one set of agents, without a significant loss in performance, even when dealing with practical navigation challenges and imperfect scene knowledge.

VIII. Conclusion

In this letter, we presented a danger estimation framework which is adaptable to different environmental settings. Our results shows its potential in enhancing team performance in SaR missions. In the future, we would like to investigate ways to reject redundant collected images, which could potentially reduce the computational overhead during online operations. In order to study a more realistic SaR scenario, we would also like to apply a dynamic Bayesian model for estimating danger. We also believe that replacing DISC images with realistic images from movies can be key in bridging the performance gap between our simulations and real-world application. Furthermore, we would like to tailor our approach to SaR missions with human-robot teams, communicating succinct and reliable information, as firefighters do in the real world.

Acknowledgment

The author would like to thank Asst. Chief Tom Basher (City of Ithaca Fire Dept.) and Chief George Tamborelle (Cayuga Heights Fire Dept.) for their valuable insights on real-world SaR missions.

References

[1] J. Casper and R. R. Murphy, "Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center," *IEEE Trans. Syst., Man, Cybern., Part B (Cybernetics)*, vol. 33, no. 3, pp. 367–385, Jun. 2003.
[2] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Auton. Robots*, vol. 15, no. 2, pp. 111–127, 2003.
[3] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *Inter J. Robot Res*, vol. 21, no. 8, pp. 735–758, 2002.
[4] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 13–23.
[5] H.-G. Jeon, S. Im, B.-U. Lee, D.-G. Choi, M. Hebert, and I. S. Kweon, "Disc: A large-scale virtual dataset for simulating disaster scenarios." in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 187–194.
[6] G. Hollinger, S. Singh, J. Djugash, and A. Kehagias, "Efficient multi-robot search for a moving target," *Inter J. Robot Res*, vol. 28, no. 2, pp. 201–219, 2009.
[7] B. Arruda Asfora, J. Banfi, and M. Campbell, "Mixed-integer linear programming models for multi-robot non-adversarial search," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6805–6812, Oct. 2020.
[8] J. P. Queralta *et al.*, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *IEEE Access*, vol. 8, pp. 191617–191643, 2020.
[9] F. Yazdani, M. Scheutz, and M. Beetz, "Cognition-enabled task interpretation for human-robot teams in a simulation-based search and rescue mission," in *Proc. 16th Conf. Auton. Agents MultiAgent Syst.*, 2017, pp. 1772–1774.
[10] X. Sun, Y. Zhang, and J. Chen, "High-level smart decision making of a robot based on ontology in a search and rescue scenario," *Future Internet*, vol. 11, no. 11, 2019, Art. no. 230, doi: 10.3390/fi11110230.
[11] K.-H. Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked cross attention for image-text matching," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 201–216.
[12] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30 k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2641–2649.
[13] S. Antol *et al.*, "Vqa: Visual question answering," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2425–2433.
[14] R. Zellers, Y. Bisk, A. Farhadi, and Y. Choi, "From recognition to cognition: Visual commonsense reasoning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6720–6731.
[15] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Trans. Assoc. Comput. Linguistics*, vol. 2, pp. 67–78, 2014.
[16] R. Yehoshua, N. Agmon, and G. A. Kaminka, "Robotic adversarial coverage of known environments," *Inter J. Robot Res*, vol. 35, no. 12, pp. 1419–1444, 2016.
[17] J. Banfi, V. Shree, and M. Campbell, "Planning high-level paths in hostile, dynamic, and uncertain environments," *J. Artif. Intell. Res.*, vol. 69, pp. 297–342, 2020.
[18] P. Anderson *et al.*, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2018, pp. 6077–6086.
[19] J. Lu, V. Goswami, M. Rohrbach, D. Parikh, and S. Lee, "12-in-1: Multi-task vision and language representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10437–10446.
[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2016, pp. 770–778.
[21] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, 2000, pp. 142–149.
[22] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.
[23] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Proc. Int. Conf. Intell. Robots Syst.*, 2004, vol. 3, pp. 2149–2154.
[24] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.
[25] Gurobi Optimization, LLC, "Gurobi - The Fastest Solver," 2019. [Online]. Available: www.gurobi.com
[26] M. Quigley *et al.*, "ROS: An open-source robot operating system," in *Proc. IEEE Int. Conf. Robot. Autom. Workshop Open Source Softw.*, Kobe, Japan, 2009.