

UC Santa Barbara

UC Santa Barbara Electronic Theses and Dissertations

Title

Optimization of Human Collaboration and Visual Scene Understanding in Robotic Operations

Permalink

<https://escholarship.org/uc/item/2vh166q2>

Author

Cai, Hong

Publication Date

2020

Peer reviewed|Thesis/dissertation

University of California
Santa Barbara

Optimization of Human Collaboration and Visual Scene Understanding in Robotic Operations

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Electrical and Computer Engineering

by

Hong Cai

Committee in charge:

Professor Yasamin Mostofi, Chair
Professor Francesco Bullo
Professor Miguel Eckstein
Professor Pradeep Sen

December 2020

The Dissertation of Hong Cai is approved.

Professor Francesco Bullo

Professor Miguel Eckstein

Professor Pradeep Sen

Professor Yasamin Mostofi, Committee Chair

September 2020

Optimization of Human Collaboration and
Visual Scene Understanding in Robotic Operations

Copyright © 2020

by

Hong Cai

To Dad, Mom, and Yuanbo

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my advisor Prof. Yasamin Mostofi for her continuous support during my Ph.D. study. Her guidance has helped me immensely in my research work. I have learned a lot from her about doing scientific research, from identifying important research problems and being rigorous during the research process to how to clearly organize and present research ideas and results.

Besides my advisor, I would like to thank the rest of my dissertation committee: Prof. Francesco Bullo, Prof. Miguel Eckstein, and Prof. Pradeep Sen, for their valuable and insightful comments.

I would also like to thank my labmates here at UCSB: Yuan, Arjun, Saandeep, Lucas, Chitra, Belal, Ekta, Mert, Anurag, and Winston. I have truly enjoyed working (and not working) with them in the lab.

Last but not least, I would like to thank my parents and Yuanbo, for all their love and support.

Curriculum Vitæ

Hong Cai

Education

2020	Ph.D. in Electrical and Computer Engineering (Expected), University of California, Santa Barbara.
2015	M.S. in Electrical and Computer Engineering, University of California, Santa Barbara.
2013	B.Eng in Electronic and Computer Engineering, Hong Kong University of Science and Technology.

Publications

(* indicates equal contribution)

Book Chapter:

- H. Cai and Y. Mostofi, “When Human Visual Performance is Imperfect – How to Optimize the Collaboration between One Human Operator and Multiple Field Robots,” *Trends in Control and Decision-Making for Human-Robot Collaboration Systems*, Springer, 2017.

Journal:

- H. Cai and Y. Mostofi, “Co-Optimization of Motion, Communication, and Sensing via Monte Carlo Tree Search,” *IEEE Transactions on Wireless Communications* (under review).
- H. Cai*, B. Korany*, C. R. Karanam*, and Y. Mostofi, “Teaching RF to Sense without RF Training Measurements,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (under review).
- B. Korany, H. Cai, and Y. Mostofi, “Multiple People Identification Through Walls Using Off-The-Shelf WiFi,” *IEEE Internet of Things Journal* (under review).
- H. Cai and Y. Mostofi, “Exploiting Object Similarity for Robotic Visual Recognition,” *IEEE Transactions on Robotics*, 2020.
- H. Cai and Y. Mostofi, “Human-Robot Collaborative Site Inspection under Resource Constraints,” *IEEE Transactions on Robotics*, 2019.
- U. Ali, H. Cai, Y. Mostofi, and Y. Wardi, “Motion-Communication Co-optimization with Cooperative Load Transfer in Mobile Robotics: An Optimal Control Perspective,” *IEEE Transactions on Control of Network Systems*, 2019.

Conference:

- B. Korany*, C. R. Karanam*, H. Cai*, and Y. Mostofi, “XModal-ID: Using WiFi for Through-Wall Person Identification from Candidate Video Footage,” *ACM International Conference on Mobile Computing and Networking*, 2019.
- E. Prashnani*, H. Cai*, Y. Mostofi, and P. Sen, “PieAPP: Perceptual Image-Error Assessment through Pairwise Preference,” *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- H. Cai and Y. Mostofi, “Asking for Help with the Right Question by Predicting Human Visual Performance,” *Robotics: Science and Systems*, 2016.
- H. Cai and Y. Mostofi, “A Human-Robot Collaborative Traveling Salesman Problem: Robotic Site Inspection with Human Assistance,” *American Control Conference*, 2016.
- U. Ali, H. Cai, Y. Mostofi, and Y. Wardi, “Motion and Communication Co-Optimization with Path Planning and Online Channel Prediction,” *American Control Conference*, 2016.
- H. Cai and Y. Mostofi, “To Ask or Not to Ask: A Foundation for the Optimization of Human-Robot Collaborations,” *American Control Conference*, 2015.

Abstract

Optimization of Human Collaboration and Visual Scene Understanding in Robotic Operations

by

Hong Cai

In this dissertation, we are interested in the optimization of mobile robotic operations that involve visual sensing and recognition. First, we show how to properly incorporate human collaboration into the robot’s field operation. More specifically, as human visual performance is not perfect and depends on the visual input, we propose a Deep Convolutional Neural Network (DCNN)-based approach to predict human performance and further show how the robot can utilize the prediction to optimally query a remote human operator. We then consider a generic robotic operation, and incorporate the predicted human performance into the co-optimization of the robot’s field sensing, motion planning, and communication, while considering the imperfect sensing quality, the realistic communication conditions, and the limited onboard energy. We pose the co-optimization as a Multiple-Choice Multidimensional Knapsack Problem, for which we propose a Linear Program-based efficient near-optimal solution, and mathematically characterize the optimality gap.

In the second part, we focus on robotic visual scene understanding under poor sensing conditions. We show that while the robot may have low confidence in classifying some objects in the environment, its onboard DCNN classifier can still provide useful information about these objects and assess if they belong to the same class. More specifically, we show that the correlation coefficient of the DCNN feature vectors of two object images carries robust information on their similarity, even though the individual sensing and

classification quality may be low. We then build a Correlation-based Markov Random Field to capture such similarity information for joint object labeling, which significantly improves the robots classification accuracy, without additional training, and further show how the robot can optimize its path and human query accordingly. This gives the robot a new way to optimally decide which object sites to move close to for better sensing and for which objects to ask for human help, which considerably improves the overall classification.

Finally, we explicitly consider the cost of communication and focus on the interplay of sensing, motion, and communication in the optimization of robotic operations. We consider the case where the robot navigates from a start position to a destination and needs to sense some sites in the field. The robot collects data when sensing each site and needs to transmit all the collected data to a remote station by the end of its trip. Our goal is to minimize the total motion and communication energy cost by co-optimizing the robot’s path, its data transmission along the path, and its sensing decisions. We propose a novel approach to solve this challenging problem by formulating a specially-designed Markov Decision Process (MDP) and utilizing Monte Carlo Tree Search (MCTS) to efficiently and optimally solve it. We theoretically prove the convergence of our approach, characterize its convergence speed, and show key properties of the optimum solution.

In order to validate our proposed methodologies, we perform extensive evaluation for each of the aforementioned parts, through realistic simulation studies and/or real-world robotic experiments. Our results demonstrate the efficacy and performance of our proposed approaches.

Contents

Curriculum Vitae	vi
Abstract	viii
List of Figures	xii
List of Tables	xviii
1 Introduction	1
1.1 Prediction of Human Visual Performance	4
1.2 Optimizing Human-Robot Collaboration with Human Performance Prediction	8
1.3 Exploiting Object Similarity for Robotic Visual Recognition	12
1.4 Co-Optimization of Motion, Communication, and Sensing via Monte Carlo Tree Search	16
2 Preliminaries	21
2.1 Communication Energy Model and Prediction in Realistic Communication Environments [1]	21
2.2 Robot Motion Energy Model [2]	23
3 Prediction of Human Visual Performance	24
3.1 Assessing Human Visual Performance on Noisy Images	25
3.2 Predicting Human Visual Performance Using Deep Learning	31
3.3 Robot Decision Optimization	37
3.4 Robot Vision	41
3.5 Performance Evaluation	41
4 Optimizing Human-Robot Collaboration with Human Performance Prediction	50
4.1 Optimizing Human-Robot Collaboration under Total Query Constraints .	52
4.2 Optimizing Human-Robot Collaboration under Total Energy Constraints	71

5	Exploiting Object Similarity for Robotic Visual Recognition	87
5.1	Correlation-based Markov Random Field	89
5.2	Optimization of Querying and Path Planning	97
5.3	Performance Evaluation on Large COCO-based Test Set	100
5.4	Performance Evaluation on a Large-Scale Drone Imagery Dataset	105
5.5	Performance Evaluation on a Large Indoor Scene Dataset	110
5.6	Robotic Experiments	112
5.7	Discussion	124
6	Co-Optimization of Motion, Communication, and Sensing via Monte Carlo Tree Search	128
6.1	Preliminaries	129
6.2	Co-Optimization Problem Formulation	134
6.3	Solving the Co-Optimization Problem via Monte Carlo Tree Search	138
6.4	Theoretical Analysis	143
6.5	Simulation Experiments	152
7	Conclusions and Future Work	158
7.1	Prediction of Human Visual Performance	159
7.2	Optimization of Human-Robot Collaboration with Human Performance Prediction	160
7.3	Exploiting Object Similarity for Robotic Visual Recognition	161
7.4	Co-Optimization of Motion, Communication, and Sensing via Monte Carlo Tree Search	162
	Appendices	163
A		164
A.1	Multiple-Choice Multidimensional Knapsack Problems	164
B		166
B.1	Constructed Image Classification Dataset	166
B.2	Descriptions of Existing Methods Included in the Performance Comparisons	166
B.3	Region-Of-Interest (ROI) Extraction Algorithm	168
B.4	Background Rejection Classifier	168
	Bibliography	170

List of Figures

1.1	A robotic surveillance task on our campus that involves finding the person in each of the 4 sites, based on imagery inputs. The robot has a limited access to a remote operator to ask for help and needs help with Sites 2 and 3. However, human performance is not perfect and thus the human cannot help with Site 2 if asked. If the robot could predict human visual performance for a given sensory input, it could optimally ask for help with Site 3 and move to Site 2 for more sensing.	5
1.2	The robot travels along a pre-defined path to inspect sites nearby. For each site, the robot decides how much it should deviate from the main path to move closer to the site for better sensing. It also needs to decide whether to ask the remote operator for help with target classification for this site. Human visual performance, however, is not perfect, and can vary depending on the sensing quality.	9
1.3	A robot is tasked with classifying objects on our campus. It may have low confidence in some of its initial classifications. Given a limited human query budget and a motion budget, the robot then needs to decide which object locations it should visit to sense better, and which object images it should ask a remote human operator to classify, in order to improve its overall classification performance. By using the correlation coefficient of the feature vectors from a trained DCNN classifier, the robot can robustly capture image similarities (e.g., objects 2 and 3 belong to the same class), even when each object is individually misclassified, which has a significant implication for its field decision-making and joint labeling, as we shall show in this chapter.	13
1.4	(Left) Example of the robotic task scenario considered in this chapter. (Right) High-level description of the co-optimization problem.	17
3.1	Gray-scale images of lion, leopard, cat, and tiger used in our study. . . .	26
3.2	A sample corrupted image (lion) with the noise variance of 3.	26
3.3	Performance of human and robot in classifying the noisy targets of Fig. 3.1.	27

3.4	An example of the optimum decisions with 2,000 sites, 500 questions and an energy budget of 25% of the total energy needed to visit all the sites.	30
3.5	Flow diagram of training a convolutional neural network to predict human performance probabilistically.	33
3.6	Comparison of the true and predicted probability of successful human performance for (top-left) an easy non-dark image, (top-right) a hard non-dark image, (bottom-left) an easy dark image and (bottom-right) a hard dark image. Readers are referred to the color pdf for better viewing. . . .	36
3.7	The CDF of NMSEs of validation images.	36
3.8	Flow diagram of robot's decision making process.	40
3.9	Initial sensing images of 3 sites at crossroad 1 and the image taken after moving to Site 1 for further sensing based on our approach. Readers are referred to the color pdf for all the experimental results. The visual difficulty of the real-size images is pretty consistent with that of the smaller versions when viewed in color.	44
3.10	Initial sensing images of 3 sites at crossroad 2 and the image taken after moving to Site 1 for further sensing based on our approach.	45
3.11	Initial sensing images of 4 sites at crossroad 3 and the image taken after moving to Site 1 for further sensing based on our approach.	46
3.12	Initial sensing images of 3 sites at crossroad 4 and the image taken after moving to Site 1 for further sensing based on our approach.	47
4.1	The robot travels along a pre-defined path to inspect sites nearby. For each site, the robot decides how much it should deviate from the main path to move closer to the site for better sensing. It also needs to decide whether to ask the remote operator for help with target classification for this site. Human visual performance, however, is not perfect, and can vary depending on the sensing quality.	51
4.2	Computation time of solving problem (4.2) and obtaining its LP-based solution for 10 sites w.r.t. the number of motion decisions for each site. The computation time is plotted in log scale.	67
4.3	The robot travels along a pre-defined main path to inspect sites near the path. Along the path, the robot's communication quality to the remote operator is subject to a space-varying wireless channel power from downtown San Francisco [3].	78
4.4	The solid red curve shows the performance of our proposed LP-based solution, where the robot has a limited total combined motion and communication energy budget. The dashed blue curve shows the performance of the benchmark. The dashed green curve shows the upper bound performance of solving problem (4.8) with perfect channel knowledge.	82

4.5	The performance of our proposed energy-aware collaborative human-robot site inspection – First row: real channel power along the main path (from downtown San Francisco). Second row: predicted channel power based on 5% prior channel measurements. Third row: the height of the bars indicates the sensing difficulty level, with 0 being the easiest and 2 being the hardest. Fourth row: the height of the bars indicates how much the robot deviates from the path to sense the site. Fifth row: it shows whether the robot queries the human, with 1 indicating a query. The sites are indexed from 1 to 10 with the leftmost being site 1. The energy budget is 30% of what is needed to both reach the site locations and query the human for all the sites.	83
4.6	Planning result in a 2D wireless environment. Brighter colors (higher values) on the channel map indicate better channel qualities. In this wireless environment, the channel power ranges from -5.40 dBm to -123.06 dBm. For a better visualization, we assign a light yellow color to locations with a channel power greater than -80 dBm and a dark blue color to locations with a channel power less than -100 dBm. The horizontal grey curve represents the pre-defined main path and the orange triangles are the points where the robot deviates from the main path to perform further sensing. The house icons indicate the site locations. For each site, the yellow dashed line indicates the extent of the deviation, and the presence of a human icon indicates that the robot will query the human operator. Readers are referred to the color PDF for optimal viewing.	85
5.1	A robot is tasked with classifying objects on our campus. It may have low confidence in some of its initial classifications. Given a limited human query budget and a motion budget, the robot then needs to decide which object locations it should visit to sense better, and which object images it should ask a remote human operator to classify, in order to improve its overall classification performance. By using the correlation coefficient of the feature vectors from a trained DCNN classifier, the robot can robustly capture image similarities (e.g., objects 2 and 3 belong to the same class), which has a significant implication for its field decision-making and joint labeling, as we show in this chapter. Readers are referred to the color pdf to better view the images in this chapter.	88

5.2	(a) In each CDF-PDF figure pair of a DCNN classifier (e.g., the two left-most figures of AlexNet), the left figure shows the CDFs of the feature correlation of same-class (blue solid) and different-class (red dashed) pairs, and the right figure shows the PDFs of the feature correlation of same-class and different-class pairs. (b) CDFs and PDFs of the feature correlation after degrading the images by low resolution and low illumination. (c) CDFs and PDFs of the feature correlation with at least one misclassification in each image pair.	91
5.3	Robustness of our metric to visual variations – visualization of the feature correlation among 10 random bicycle images. Two images are connected if their AlexNet-based feature correlation is above the threshold. The number of edges in this graph is 55.56% of that of a 10-node complete graph, indicating that many of the pairs are correctly declared as same-class despite drastic visual differences.	94
5.4	Joint labeling and query selection performance (no motion) on the COCO-based test set. The query budget is given as a fraction of the total number of nodes.	103
5.5	Query selection and path planning performance on the COCO-based test set. The robot is given a query budget equal to 5% of the total number of nodes and the motion budget ranges from 0 m to 20 m.	104
5.6	Joint labeling and query selection performance (no motion) on the VisDrone dataset, for the same-flight scenario. The query budget is given as a fraction of the total number of nodes.	107
5.7	Joint labeling and query selection performance (no motion) on the VisDrone dataset, for the multi-flight scenario. The query budget is given as a fraction of the total number of nodes.	107
5.8	Query selection and path planning performance on the VisDrone dataset. The robot is given a query budget equal to 5% of the total number of nodes and the motion budget ranges from 0 m to 20 m.	109
5.9	Joint labeling and query selection performance (no motion) on the NYU-v2 dataset. The query budget is given as a fraction of the total number of nodes.	111
5.10	Query selection and path planning performance on the NYU-v2 dataset. The robot is given a query budget equal to 5% of the total number of nodes and the motion budget ranges from 0 m to 20 m.	111
5.11	The diagram shows the robot’s steps in an experiment using our proposed approach. The dashed line between the two “CoMRF” blocks indicates that they are the same MRF, based on the initial classification and feature extraction. The top CoMRF is then further updated with the image labels obtained from visit or query.	114

5.12	Object images obtained by the robot in Exp. 1 on our campus. A line between two images indicates that there is an edge between them in the corresponding CoMRF. See the color pdf to better view all the experiment images.	115
5.13	Performance (classification accuracy) of CoMRF and the independent approach w.r.t. the number of allowed queries in Exp. 1 (shown in Fig. 5.12).	115
5.14	Object images obtained by the robot in Exp. 2 on our campus. A line between two images indicates that there is an edge between them in the corresponding CoMRF.	116
5.15	Performance (classification accuracy) of CoMRF and the independent approach w.r.t. the number of allowed queries in Exp. 2 (shown in Fig. 5.14).	116
5.16	Object images obtained by the robot in Exp. 3 on our campus. A line between two images indicates that there is an edge between them in the corresponding CoMRF.	118
5.17	Performance (classification accuracy) of CoMRF and the independent approach w.r.t. the number of allowed queries in Exp. 3 (shown in Fig. 5.16).	118
5.18	Object images obtained by the robot in Exp. 4 on our campus. A line between two images indicates that there is an edge between them in the corresponding CoMRF.	120
5.19	Blue circles indicate the object locations and the red square indicates the robot's initial position in Exp. 4 of Fig. 5.18. The green dashed line indicates an edge in the CoMRF. (a)-(b) show the respective path planning results of the independent approach and CoMRF-Opt, where the solid black line indicates the sensing tour.	120
5.20	Object images obtained by the robot in Exp. 5 on our campus. A line between two images indicates that there is an edge between them in the corresponding CoMRF.	121
5.21	Blue circles indicate the object locations and the red square indicates the robot's initial position in Exp. 5 of Fig. 5.20. The green dashed line indicates an edge in the CoMRF. (a)-(b) show the respective path planning results of the independent approach and CoMRF-Opt, where the solid black line indicates the sensing tour and the purple diamond indicates the queried object.	121
5.22	Object images obtained by the robot in Exp. 6 on our campus. A line between two images indicates that there is an edge between them in the corresponding CoMRF.	122
5.23	Blue circles indicate the object locations and the red square indicates the robot's initial position in Exp. 6 of Fig. 5.22. The green dashed line indicates an edge in the CoMRF. (a)-(b) show the respective path planning results of the independent approach and CoMRF-Opt, where the solid black line indicates the sensing tour and the purple diamond indicates the queried object.	122

5.24	Performance (classification accuracy) of CoMRF and the independent approach w.r.t. the number of allowed queries, when jointly classifying all the experimental images (shown in Figs. 5.12, 5.14, 5.16, 5.18, 5.20, and 5.22).	123
5.25	Each figure shows the CDFs of the feature correlation of same-class (blue solid) and different-class (red dashed) pairs, based on the Inception-v3 and ResNet-101 classifiers, respectively.	125
6.1	(Left) Example of the robotic task scenario considered in this chapter. (Right) High-level description of the co-optimization problem.	129
6.2	Paths from solving problem (6.5) with three sites to sense, by using our proposed approach (yellow) and the benchmark (green). The yellow dots and solid curve represent the waypoints and the path, respectively, as given by our proposed approach. The green crosses and dashed curve represent the waypoints and the path given by the benchmark method. The circle and square indicate the start and final positions, respectively. The white triangles indicate the site locations and the white circles indicate the sensing ranges for the sites. The colormap indicates the true channel power over this environment, where brighter (darker) colors indicate higher (lower) channel qualities. See the color pdf for optimal viewing.	153
6.3	The optimal transmission spectral efficiencies from solving problem (6.5) with three sites to sense, by using our proposed approach. The first and second rows show the actual and the predicted channel powers at the waypoints, respectively. The third row shows the optimum spectral efficiencies for the waypoints.	154
6.4	The resulting path when using our proposed approach to solve problem (6.5) for the case of no sensing. The yellow curve shows the path and the yellow dots indicate the waypoints. The circle and square indicate the starting and final positions, respectively. The colormap indicates the true channel power over this environment, where brighter (darker) colors indicate higher (lower) channel qualities. See the color pdf for optimal viewing.	156
6.5	The optimum transmission spectral efficiencies from solving problem (6.5) when there is no sensing, by using our proposed approach. The first and second rows show the actual and the predicted channel powers at the waypoints, respectively. The third row shows the optimum spectral efficiencies for the waypoints.	157

List of Tables

3.1	Performance of our human predictor – NMSEs over different subsets of images in the validation set.	34
3.2	Comparison of the performance of our predictor with two existing predictors. N/A indicates that the evaluation is not available. The performance of our predictor is evaluated over all non-dark validation images, which best match the characteristics of the images used in [4, 5].	37
3.3	Performance summary at campus crossroad 1.	44
3.4	Performance summary at campus crossroad 2.	45
3.5	Performance summary at campus crossroad 3.	46
3.6	Case of 15 sites and 5 allowed queries. The table shows robot and human performance for each site, as well as the prediction of human performance.	48
4.1	Motion energy saving by our proposed approach as compared to the benchmark. In this example, the robot is given 6 allowed queries.	71
4.2	Saving of queries by our proposed approach as compared to the benchmark. In this example, the motion energy budget is 50% of what is needed to visit all the site locations.	71
4.3	Energy saving of our proposed approach as compared to the benchmark, for the case where the robot has a limited total motion and communication energy budget.	82
5.1	Correlation thresholds for the three DCNN classifiers. The third and fourth columns show the probability of false correlation (different-class declared as same-class) and the probability of a same-class pair having a correlation above the threshold, respectively.	93
5.2	Performance comparison of joint labeling methods (case of no query or motion). It can be seen that our proposed joint labeling method improves the base classifier by 0.195, on average, in terms of classification accuracy. On the other hand, Cao et al. and Hayder et al. perform similar or worse as compared to the initial classification.	101

5.3	Average classification performance given different number of classes present (N_p) in a test case. The first column shows the number of classes present in a test case. The second column shows the average number of objects per class in a test case for different N_p values. The remaining columns show the average classification accuracy of the benchmark of making independent decisions and our proposed CoMRF-based approach, given different N_p values, for the base classifiers of AlexNet, MobileNet-v2, and ResNet-18, respectively.	108
6.1	Average energy costs by using our proposed approach and the benchmark method, over 50 random problem instances. The second, third, and fourth columns show the communication, motion, and total energy costs (in Joules), respectively. The last row shows the total energy saving by using our proposed approach, as compared to the benchmark.	154
B.1	List of object classes in our dataset, along with the numbers of images in the training/validation/test sets.	167

Chapter 1

Introduction

Recent years have seen great advances in robotics, especially in the areas of vision and learning. Thanks to these technological developments, robots are now capable of performing a wide range of tasks, such as surveillance, search and rescue, and exploration of new environments. However, due to the various challenges in realistic robotic operations, the robot can still encounter uncertainties in its visual sensing and recognition in reality. For instance, consider a visual sensing operation where the field mobile robot is tasked with recognizing targets of interest in the environment. Due to the noisy and uncertain sensing data (e.g., due to far sensing distance, poor lighting conditions, etc.), the robot may not be able to recognize some of these targets.

In order to improve its visual recognition performance, the robot can send the visual data (e.g., images) to a remote human operator to ask for help, since humans have superior visual capabilities. Human visual performance, however, is not perfect, as has been heavily acknowledged in cognitive psychology [6]. In addition, human visual performance can vary drastically depending on the sensory/image input, as we have established in our work [7, 8]. As such, it is key to take the imperfect human visual performance into account in the robot's decision-making. In order to do this, we need to predict human visual performance given a visual input. This would then allow the robot to ask for human help only when the human can help. Given the prediction of human performance,

the robot should then optimize its field actions accordingly, in terms of sensing, motion, and query/communication. In other words, the robot should optimally decide for which visual inputs it can query the remote human operator, for which locations it needs to move closer to acquire higher-quality sensing data, and for which visual data it can rely on its onboard vision system.

Another way for the robot to improve its visual recognition performance is to infer object similarity in its field of view, even when the individual object sensing quality is poor. In order to do this, the robot needs to robustly extract such object similarity information from its Deep Convolutional Neural Network (DCNN) output, even when the DCNN misclassifies the objects, and properly utilize this information for its field sensing and decision-making.

In addition to the visual sensing and recognition aspect, there are other realistic factors that need to be considered in mobile robotic operations, such as the realistic wireless communication channel and the robot's limited onboard energy. It is important to take these factors into account, when optimizing the robot's field actions, in terms of sensing, motion, and communication. For instance, the quality of the wireless communication channel varies spatially, due to the wireless fading effects. The robot needs to take this into account and optimally select the locations where it should transmit its onboard sensing data to a remote human operator (or a remote station), such that the transmission quality is acceptable and the required transmission energy cost is not excessive.

In this dissertation, we will explore all these aspects through the following topics:

1. *Prediction of Human Visual Performance:* We show how to use a DCNN to accurately predict the probability that a person can successfully perform a given visual task. We then demonstrate how the robot can utilize such prediction information to effectively collaborate with a human operator in visual recognition tasks, via

several real-world robotic experiments.

2. *Optimizing Human-Robot Collaboration with Human Performance Prediction:* We incorporate the human performance prediction into the optimization of the robot's field decisions, in terms of sensing, motion, and querying/communication, while considering the robot's sensing quality, the non-ideal communication link to the human operator, and the robot's limited onboard energy. In this part, we extensively study the theoretical properties of such complex robotic decision optimization problems and derive computationally efficient near-optimal solutions.
3. *Exploiting Object Similarity for Robotic Visual Recognition:* We show that even when the robot's onboard DCNN classifier cannot properly classify the objects in the field, its output still contains useful similarity information about the objects (e.g., what objects belong to the same class). We show how to robustly extract such similarity information from the DCNN output and properly exploit it to improve the classification accuracy. When the robot is given a number of queries and a motion budget, we then show how we can utilize the extracted visual similarity information to optimize the robot's field actions, in terms of sensing, motion, and querying. This allows the robot to derive more efficient actions and significantly improve its visual classification accuracy.
4. *Co-Optimization of Motion, Communication, and Sensing via Monte Carlo Tree Search:* In this part, we consider a more general sensing model and focus on the comprehensive co-optimization of the robot's sensing, motion, and communication, in realistic wireless communication environments. In order to solve the resulting highly-complex optimization problem, we propose a specially-formulated Markov Decision Process (MDP) and utilize Monte Carlo tree search (MCTS) to efficiently and optimally solve it.

We next discuss in more detail each of the four aforementioned topics, where we place our work in context by reviewing the relevant literature and summarize our contributions.

1.1 Prediction of Human Visual Performance

In Chapter 3, we consider robotic visual surveillance tasks, where the robot has a limited number of chances to ask a remote human operator for help with visual recognition. In order for the robot to properly utilize the human queries, we propose a Deep Convolutional Neural Network (DCNN)-based approach to predict human visual performance for any given image, as human visual performance is not perfect and varies depending on the quality of the visual input. This enables the robot to ask the right questions, only querying the human operator with the images for which humans have a high chance of successful visual recognition. Then, the robot will use the feedback from our predictor to optimize its human collaboration and further sensing of the field.

Fig. 1.1 demonstrates a real example of this. A robot is given a visual perception task that involves finding the human at each of the four shown sites, based on the onboard camera inputs. The robot has a limited access to a remote human operator to ask for help with this task. The figure shows the images taken by the robot after its initial sensing of each site. Equipped with a state-of-the-art vision algorithm, the robot has no problem recognizing the humans at Sites 1 and 4 itself. However, the robot's vision fails for Sites 2 and 3. On the other hand, the person at Site 3 can be easily detected by a human, while it is also hard for humans to spot the person at Site 2 based on the current sensory input. Thus, if the robot assumes that human is perfect and queries the human operator with the second image, it is highly likely that the operator will fail to find the human in the image. This real-world example thus shows that it is important for the robot to predict human visual performance given the visual inputs, in order to optimally

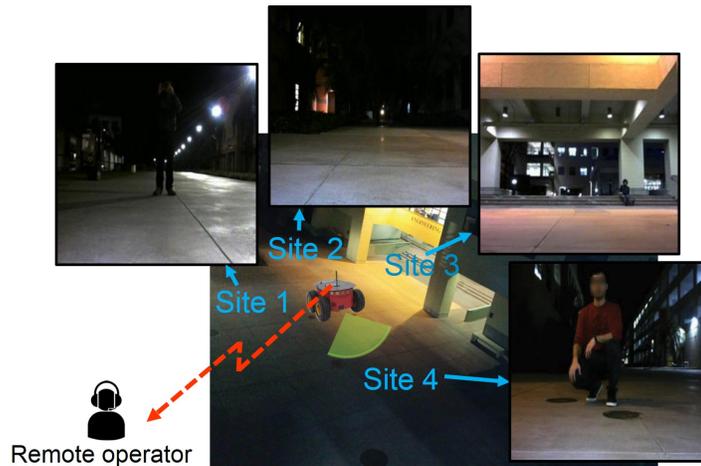


Figure 1.1: A robotic surveillance task on our campus that involves finding the person in each of the 4 sites, based on imagery inputs. The robot has a limited access to a remote operator to ask for help and needs help with Sites 2 and 3. However, human performance is not perfect and thus the human cannot help with Site 2 if asked. If the robot could predict human visual performance for a given sensory input, it could optimally ask for help with Site 3 and move to Site 2 for more sensing.

determine its actions.

1.1.1 Related Work

There is a great body of work on different aspects of human-robot collaboration [9–13]. More related to this chapter are those papers that focus on robots asking human for help. For instance, [14] shows how robots can recover from difficult states or failures by asking for help. In [15–17], a robot learns from human demonstration and correction, while a robot performs object detection and recognition with human inputs in [18–21]. In computer vision, a number of work have focused on designing human-machine interfaces that allow the vision algorithm to ask for human help when it encounters difficulties [22–25]. In most of these papers on asking for help, however, human is assumed perfect in task accomplishment. As can be seen in Fig. 1.1, however, human visual performance is not perfect and depends on the sensory input.

A number of studies have taken imperfect human performance into account for non-robotic vision applications. In [22], for instance, authors propose a collaborative vision task inspired by the 20-question game. [24] proposes a collaborative annotation system in which human and machine collaboratively label the objects in images. In these papers, however, it is assumed that human performance is task dependent but invariant to the sensory inputs. In other words, for a specific visual task, the probability of human’s task accomplishment is constant (less than 1), but independent of the input image. While a good step towards considering imperfect human performance, human visual performance can largely vary for a given visual task, depending on the sensory input, as shown in Fig. 1.1.

A few studies have attempted to estimate human visual performance based on a given input image. Johnson criteria is one of the first attempts along this line, where human visual performance is predicted based on the number of line pairs of display resolution occupied by the target on the screen [26]. In cognitive psychology, it is heavily acknowledged that human visual system is not perfect [6, 27, 28], motivating work that attempt to understand how different image features affect human performance [29]. The goal of these studies, however, is not predicting human performance for a given image but understanding a certain feature’s impact on human performance.

In [4, 5], the authors utilize machine learning to predict the probability that a driver is able to detect a pedestrian at a glance, based on hand-crafted image features, such as the size and position of the pedestrian in the image. In general, however, an image may have several features that can make it hard or easy for humans to perform a visual task, making identifying and hand-crafting all of them very challenging. Thus, an automated method that can predict human visual performance for any given sensory input is needed and currently lacking, which is the main motivation for this chapter. More specifically, as compared to the existing approaches, our proposed prediction methodology is fundamen-

tally different in that the relevant image features are selected in an automated manner during training, by properly utilizing DCNNs, and then used for predicting human visual performance.

1.1.2 Contributions

The main contributions of this chapter are summarized as follows:

1. We propose a DCNN-based approach that allows the robot to probabilistically predict human visual performance for a given visual input, without a need for hand-crafting any feature. We train our DCNN by gathering several human data using Amazon Mechanical Turk (MTurk) [30]. We then test the proposed predictor on a large validation set and show a considerable reduction in prediction error as compared to the state-of-the-art.
2. We show how the robot can optimize its field decisions in terms of relying on itself, asking for help, and further sensing, based on the output of the predictor.
3. We run a number of robotic surveillance experiments on our campus, which showcase the effectiveness of our approach, indicating a considerable increase in the success rate of human queries and the overall task. We further run a larger-scale evaluation in a simulation environment, based on images taken on our campus and with MTurk users acting as human operators.

1.2 Optimizing Human-Robot Collaboration with Human Performance Prediction

In Chapter 4, given the predicted human performance, we are then interested in a more comprehensive and in-depth study of incorporating the prediction into the optimization of human-robot collaboration, while also considering various factors commonly-seen in robotic operations, such as realistic wireless channel environments and limited onboard energy. More specifically, we consider a realistic, general robotic operation scenario where there is a pre-defined path, near which there are a number of sites containing targets to be classified, as shown in Fig. 1.2. As the robot traverses this path and reaches a point close to a site, it has to decide whether (and to what extent) it should incur motion energy to deviate from the main path towards the site to sense it better, and whether it should ask for human help with this site. The robot then returns to the pre-defined path after a possible deviation. This scenario captures several realistic cases of robotic missions. For example, a security surveillance drone can have a pre-defined patrol route, but may need to deviate from the route to investigate suspicious locations off the route. In another example, an indoor monitoring robot has to survey the offices down a hallway. The pre-defined route is then set along the hallway. When the robot needs to gather more sensing information about a room, it has to decide how far it should move into a room and whether it should query a remote human operator for help with this particular room.

The main goal of this chapter is then to provide an optimization framework for a generic and realistic problem of human-robot collaborative site inspection under resource constraints. More specifically, in the considered setup of this chapter, the robot's objective is to maximize its target classification performance at all the sites, under limited onboard energy constraints (including both communication and motion), with a limited access to a human operator to ask for help, and while considering the fact that human

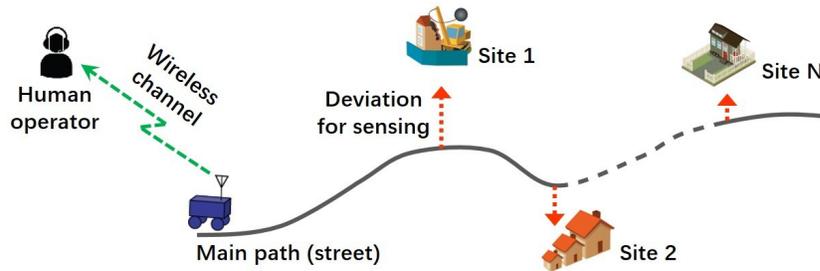


Figure 1.2: The robot travels along a pre-defined path to inspect sites nearby. For each site, the robot decides how much it should deviate from the main path to move closer to the site for better sensing. It also needs to decide whether to ask the remote operator for help with target classification for this site. Human visual performance, however, is not perfect, and can vary depending on the sensing quality.

visual performance is not perfect and varies depending on the sensory input. Thus, before deciding when to query the human operator, the robot needs to predict human visual performance over the sensory inputs. Furthermore, the robot has constraints in communication with the human operator. We consider two realistic communication constraints. In our first case, the robot is given a limited number of chances to query the human operator. This case realistically considers the impact of possible human work overload, which needs to be prevented by limiting the number of queries. Another cause of communication constraint is the quality of the wireless channel in the environment, when communicating with the human operator. In other words, the quality of the link may not be good enough to establish a reliable communication link all over the workspace, due to path loss, shadowing, and multipath fading in realistic channel environments. In our second case, the robot then needs to first predict the channel quality at unvisited locations in the environment (e.g., using the probabilistic channel prediction framework of [1]) and decide where to communicate with the human operator, given a limited energy budget that can be used towards communication and motion.

Due to the communication restrictions and the imperfect human visual performance, the robot cannot query the operator all the time. Instead, the robot should only ask

for help when human assistance can bring reasonable performance improvements (i.e., the task is doable by humans with a high probability) and when the cost of querying is reasonable. On the other hand, the robot may not have sufficient motion energy to sense the targets to the extent that it can fully rely on its own classification performance. Therefore, to achieve a good overall classification performance, it is necessary to co-optimize the robot's motion, sensing, and queries to the remote human operator, while considering the impact of limited energy, the imperfect human performance, and the realistic wireless channel environments.

1.2.1 Related Work

For visual inspection tasks, a number of papers propose human-machine interface designs that allow the vision algorithm to query the human in order to improve its performance [22–24,31,32]. In these papers, human visual performance is assumed either perfect or independent of the difficulty/quality of the individual sensory/image input. Human visual performance, however, is not perfect, as has been heavily acknowledged in cognitive psychology [6]. In addition, human visual performance can vary drastically depending on the sensory/image input, as we have established in our previous work [7,8].

Next, consider robotic field operation and decision making under resource constraints, in terms of sensing [33–35], motion planning [2,36,37], and communication [38–40], which have received considerable attention over the years. More recently, researchers have become interested in communication-aware robotics, which considers realistic wireless channel environments and addresses the co-optimization of motion, sensing, and communication, under limited resources. For instance, Ghaffarkhah et al. study robotic target tracking and coverage under wireless fading channels [41,42]. Yan et al. study the co-optimization of communication and motion in robotic operations [43]. Optimal

control-based co-optimization methods have also been proposed in [44]. However, in these work, human collaboration and the impact of imperfect human performance are not considered.

1.2.2 Contributions

The main contributions of this chapter are summarized as follows:

1. We show how to co-optimize the robot’s motion, sensing, and human queries in human-robot collaborative site inspections, for the realistic case that the human visual performance is not perfect, and under limited energy resources and communication constraints to the operator, while considering realistic wireless channel environments that experience path loss, shadowing, and multipath fading. Given probabilistic predictions of human performance and channel quality, we show how to formulate the resulting co-optimization problems as Multiple-Choice Multidimensional Knapsack Problems (MMKP) [45].
2. We then propose a Linear Program (LP)-based efficient near-optimal solution to the NP-hard MMKP, and mathematically characterize the optimality gap, confirming that it can be considerably small. We also mathematically characterize properties of the optimal solution.
3. We comprehensively validate our proposed approach with real human visual performance data that we acquired from extensive user studies on Amazon Mechanical Turk (MTurk), real wireless channel data collected from downtown San Francisco, and realistic motion and communication models. The numerical results show that our proposed approach properly co-optimizes motion, sensing, and queries, and significantly outperforms benchmark methodologies.

1.3 Exploiting Object Similarity for Robotic Visual Recognition

Consider a robotic operation where a robot is tasked with visual sensing and classification in an area, an example of which is shown in Fig. 1.3. The robot takes several object images with its camera and uses a trained DCNN classifier to classify them to a given set of classes. Due to factors such as poor sensing, low lighting and illumination, and less-seen poses/views of the objects, however, the robot’s DCNN classifier may misclassify or have low classification confidence for some of the images. If the robot can figure out which objects belong to the same class (even when misclassifying them), it can then improve its overall classification accuracy of the scene.

In Chapter 5, we show that the object similarity information can be freely and robustly inferred from the output of a trained DCNN classifier, without requiring any additional training. More specifically, we show that the Pearson correlation coefficient of the feature vectors of two object images from the output of a trained DCNN classifier carries reliable information on the similarity of the two corresponding objects (i.e., do they belong to the same class?), even if the individual classification accuracy of each object is not very high. This then allows us to design a Markov Random Field (MRF)-based joint labeling framework, where the similarity information is utilized to reduce the classification uncertainty. For instance, in Fig. 1.3, there are two images (2 and 3) of the same class (person) but in different poses. The robot’s vision initially misclassifies them to a truck and a car. However, if the robot can infer that they belong to the same class (while it cannot properly classify them), then it can correctly classify both to persons using our MRF joint labeling framework. We then show the implication of this visual understanding and correlation modeling for the robot’s field decision-making by co-optimizing its query, motion, and visual labeling.

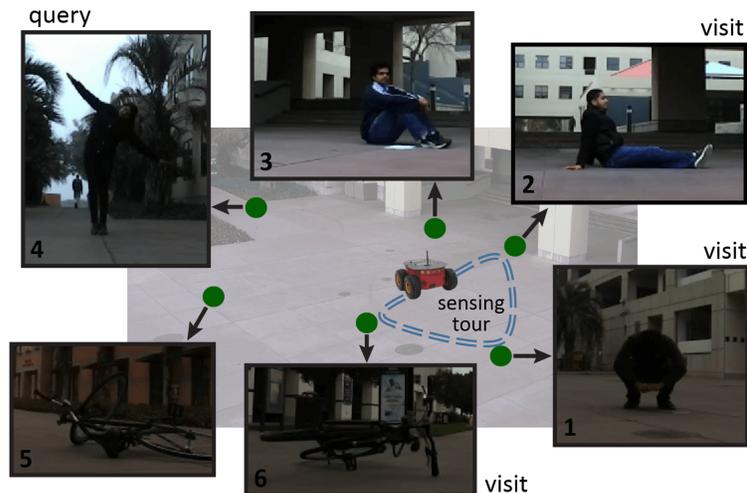


Figure 1.3: A robot is tasked with classifying objects on our campus. It may have low confidence in some of its initial classifications. Given a limited human query budget and a motion budget, the robot then needs to decide which object locations it should visit to sense better, and which object images it should ask a remote human operator to classify, in order to improve its overall classification performance. By using the correlation coefficient of the feature vectors from a trained DCNN classifier, the robot can robustly capture image similarities (e.g., objects 2 and 3 belong to the same class), even when each object is individually misclassified, which has a significant implication for its field decision-making and joint labeling, as we shall show in this chapter.

1.3.1 Related Work

In computer vision, machine learning and deep neural networks have significantly advanced the state of the art, in areas such as detection [46], segmentation [47], and DCNN architectural design [48–51]. While most research efforts have focused on making the machine better at processing individual visual inputs, relationship among a number of visual inputs can also be exploited to design better vision systems, as some recent papers show. For instance, Galleguillos et al. [52] utilize object co-occurrence and spatial context to design categorization algorithms. Torralba et al. [53] study place-object co-recognition, where the semantic consistency between an object and the current place is taken into account. In image segmentation, pixel-level relationship is utilized in the classification of each pixel [47]. Spatial-temporal correlation has also been considered in

video applications [54,55]. More related to our work are those that consider object/image similarity. Several papers have proposed DCNN-based similarity metrics, which require dedicated training to learn a similarity measure for a specific application (e.g., patch matching, image retrieval, image error assessment) [56–60]. A few papers have explored off-the-shelf DCNN features for applications such as clustering [61,62]. As for visual detection and classification, a few papers have utilized pairwise object/image similarity to improve the recognition accuracy [63–65]. However, these methods use simple hand-crafted features (e.g., color histogram), which will not work well in complex visual tasks that can involve many object classes, different poses/views of the same object class, and/or images degradations.

The robotics community has also started to exploit the robot’s understanding on the relationship among visual inputs for robotic vision applications. For instance, for object labeling, Koppula et al. [66] incorporate geometric context among objects, and Ali et al. [67] utilize the co-occurrence relationship. Ruiz et al. [68] utilize semantic knowledge to derive compatibilities among objects and rooms for joint recognition. However, these methods require additional training in order to use the contextual relations. For instance, extensive training is needed to utilize geometric relations for joint labeling in [66].

In some cases, the machine can obtain a few ground-truth labels via querying to improve its performance. Most related to this work are those that optimize the query selection based on an already-given correlation model (e.g., an MRF). In [69], Krause et al. utilize information theoretic metrics (e.g., mutual information) to select queries most beneficial to labeling the remaining un-queried instances. Recently, Wang et al. [70] use Bayesian lower bounds to optimize the selection, which outperforms the earlier information theoretic methods. Another related subject is active learning, which studies how to select labeled samples to better train a learning algorithm (e.g., [54,71,72]). The formulation of active learning, however, is different from our problem, as we do not con-

sider retraining the vision algorithm during deployment in this work. In the context of robotics, several papers have studied how to optimize the robot’s motion to acquire more information and improve its visual sensing [73–75]. These existing papers, however, do not take into account the correlation among the visual targets.

As discussed above, while several types of relationships between visual inputs have been exploited in robotic vision, object similarity has not been exploited in robotics. It is our hypothesis that the similarity between two visual inputs can be inferred robustly from the output of a trained DCNN classifier, without any additional training. This would then have a significant implication for the robot’s visual classification, and its field decision-making in terms of visual sensing, path planning, and querying, as we shall show in this chapter.

1.3.2 Contributions

The main contributions of this chapter are summarized as follows:

1. We probabilistically analyze the correlation coefficient between the features of two images from a trained DCNN classifier in an extensive study based on 180,000 image pairs from 39 classes, for 3 commonly-used state-of-the-art DCNN architectures. We show that the correlation coefficient can capture pairwise image similarity robustly, even when the images are subject to low illumination and low resolution, or are misclassified. This similarity measure comes for free from the DCNN classifier, requiring no additional training.
2. Based on the probabilistic characterization of this pairwise image similarity metric, we build a correlation-based MRF (CoMRF) for joint labeling, which allows the robot to better label the objects based on its inferred correlation structure.

Given query and motion budgets, we propose a CoMRF-based query-motion co-optimization approach to jointly plan the robot’s query and path. This allows the robot to optimally decide which objects it should visit for better sensing, and for which visual inputs it should ask for human help. As we shall see, by utilizing the proposed framework, the robot can improve its visual performance significantly, under the motion and query budgets. In other words, our proposed method significantly reduces the robot’s motion and query/communication burdens for labeling, while achieving the same task quality.

3. By using 1) a large COCO-based test set [76], 2) the challenging large-scale drone imagery dataset of VisDrone [77], and 3) the large indoor scene dataset of NYU-v2 [78], we extensively evaluate our proposed approach on joint labeling, query selection, and path planning, across a large variety of realistic scenarios. The results show that our proposed approach significantly outperforms the state of the art. We then run several real-world robotic experiments to further demonstrate the efficacy of our proposed CoMRF-based query-motion co-optimization algorithm. The results verify that our approach considerably outperforms the benchmark.

1.4 Co-Optimization of Motion, Communication, and Sensing via Monte Carlo Tree Search

In the previous chapters, we focused on the components of human collaboration and object similarity for enhancing the visual recognition performance of a mobile robot. In Chapter 6, we more explicitly bring in path planning and communication, and adopt a more general sensing model of the robot, in order to more comprehensively study the co-optimization of the robot’s motion, communication, and sensing. More specifically,

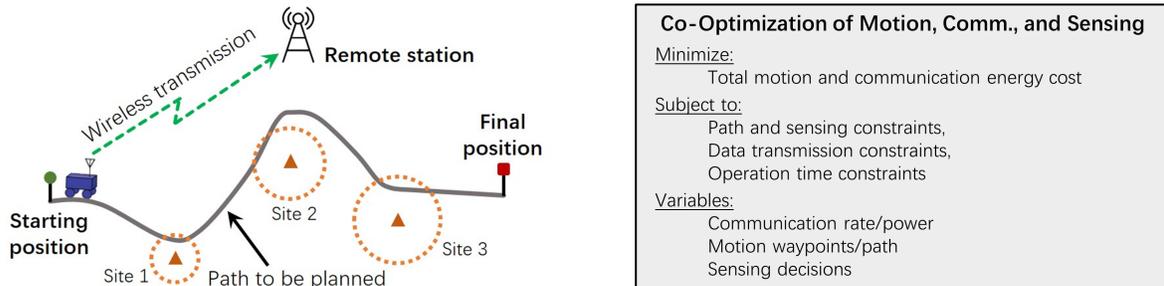


Figure 1.4: (Left) Example of the robotic task scenario considered in this chapter. (Right) High-level description of the co-optimization problem.

we are interested in a general robotic task scenario that involves planning the robot’s entire path, its data transmission along the path, as well as its sensing in the field. As illustrated in Fig. 1.4, the robot navigates from the start position to a given final position, and needs to sense a number of sites in the field. For each site, the robot must move within a certain distance to sense it (e.g., to ensure an acceptable level of sensing quality). When visiting each site, the robot collects new sensing data. The robot is required to transmit all its collected data (and possibly some initial data) to the remote station (or a remote human operator) by the end of the trip, while traversing the field. Its transmission energy cost is subject to a spatially-varying wireless communication channel that experiences fading. Our goal is to minimize the total motion and communication energy cost of this robotic operation, by optimizing the robot’s sensing and path, as well as its data transmission as it traverses the path. Fig. 1.4 (right) shows a high-level description of our co-optimization problem. This setup captures a variety of real-world robotic applications. For instance, an environment-monitoring robot needs to collect sensing data from a few places. For each place, it must move close enough to sense it. It needs to transmit all the collected data to a remote station, before getting to a given destination and under resource constraints.

In order to minimize the total motion and communication energy consumption, the co-optimization needs to take into account the locations of the sites, the distance that the

robot should move within each site for sensing, the destination location, the spatially-varying wireless channel, and the operation time budget, in order to plan the robot's entire path as well as its data transmission during the trip. This is a considerably challenging problem, which no existing approach in the literature is capable of solving. More specifically, it involves the combinatorial characteristics of path planning, the path constraints imposed by the sensing requirements, the spatially-varying wireless channel which cannot be modeled analytically, and the nonlinear transmission optimization. In order to solve this co-optimization problem, we propose a novel approach utilizing Markov Decision Process (MDP) and Monte Carlo Tree Search (MCTS), and show how it can solve this problem efficiently while providing theoretical convergence guarantees.

1.4.1 Related Work

Recent years have seen considerable research interest in the co-design of motion and communication in mobile robotics, where wireless communication models are explicitly taken into account in the optimization of mobile robotic operations [43, 44, 82–106]. While some of the earlier papers use over-simplified communication models, such as the disk/path loss model, and/or assume perfect knowledge of the channel [82–86], which can lead to poor performance in realistic wireless channel environments [88], researchers have recently started to incorporate realistic wireless fading models into various aspects of the co-optimization of motion and communication in mobile robotics, such as data relaying [91–94], cooperative transmission [95–97], robot-assisted wireless coverage [98–102], and data gathering [43, 103–105].

More related to this chapter are studies on the co-optimization of motion, communication, and sensing for data transmission from a field robot to a remote station in realistic channel environments and under resource constraints. Due to the complex spatial dy-

namics of a wireless channel and the challenges imposed by sensing and path planning, most existing work in this area optimize only a subset of the robot’s motion and transmission actions while assuming that the rest are predefined, in order to bring tractability to the problems. For instance, [106] optimizes the robot’s deviation from a predefined main path for better sensing and communication, and the robot only transmits data at the sensing spots (rather than throughout the entire path). In [43], the authors consider a motion-communication co-optimization problem with site visits. They assume that the robot has to visit the exact site locations and the route between two sites is a straight line. On the other hand, Ali et al. have recently shown how to co-optimize the entire path and the data transmission along the path, but without any sensing considerations and by using a numerical algorithm derived from optimal control, thus lacking theoretical convergence guarantees [44].

1.4.2 Contributions

The main contributions of this chapter are summarized as follows:

1. We study a complex robotic co-optimization problem, which involves sensing, data transmission, and path planning, in a realistic wireless channel environment. We propose a novel approach to solve this co-optimization problem by formulating a specially-designed MDP and utilizing MCTS to efficiently and optimally solve it. As we shall see, by considering the transmission optimization in the MDP terminal reward evaluation and the motion actions in the state transitions, we can reduce the complexity of this problem and iteratively optimize the sensing/motion and the communication parts in different stages of MCTS, which enables us to efficiently solve the original challenging co-optimization problem.
2. We mathematically characterize an upper bound on the probability that our pro-

posed approach does not result in the optimum decision. This result shows that our algorithm converges to the optimum as the number of iterations increases and provides a bound on the convergence speed. We further mathematically characterize the properties of the optimum solution.

3. Using a realistic 2D wireless channel environment, we thoroughly validate the performance of our proposed approach. We further compare our proposed approach with a benchmark method that separately optimizes motion and communication, and show that our approach significantly outperforms the benchmark. For instance, based on an evaluation on 50 problem instances, the solution given by our proposed approach uses 55% less total energy on average, as compared to the benchmark. Finally, we compare with the most related state-of-the-art work on this area.

Chapter 2

Preliminaries

In this chapter, we provide an overview on the communication energy model and the realistic wireless channel prediction framework proposed in [1], as well as a brief overview on the robot's motion energy model. These models will be used throughout the rest of this dissertation.

2.1 Communication Energy Model and Prediction in Realistic Communication Environments [1]

Consider the case where the robot adopts the commonly-used MQAM modulation for transmission. As shown in [107], the required transmission power can be well approximated by $\Gamma_T = (2^r - 1)\ln(5p_{\text{BER}})/(-1.5\Upsilon)$, where r is the spectral efficiency (i.e., the transmission rate divided by the bandwidth, in bits/s/Hz), p_{BER} is the required Bit Error Rate (BER), and Υ is the received Channel-to-Noise Ratio (CNR). During operation, the robot needs to assess its transmission power along any given unvisited path for the purpose of path planning. This, however, requires it to have a prediction of the channel at unvisited locations over the space, based on a small number of prior or online channel samples. Due to the real-world wireless propagation effects of path loss, shadowing, and

multipath fading, the CNR is best modeled as a spatial stochastic process. Then, given a small number of prior channel samples in the same environment and based on [1], the CNR (in dB) at an unvisited location q , $\Upsilon_{\text{dB}}(q)$, can be best modeled by a Gaussian random variable, with its expectation and variance given by

$$\begin{aligned}\mathbb{E}[\Upsilon_{\text{dB}}(q)] &= H_q \hat{\theta}_{ch,dB} + \Psi^T(q) \Phi^{-1}(Y - H_{Q_m} \hat{\theta}), \\ \Sigma(q) &= \hat{\alpha}_{ch,dB}^2 + \hat{\sigma}_{ch,dB}^2 - \Psi^T(q) \Phi^{-1} \Psi(q),\end{aligned}\tag{2.1}$$

where $Y = [y_1, \dots, y_m]^T$ are the m priorly-collected CNR measurements (in dB), $Q_m = [q_1, \dots, q_m]$ are the measurement locations, $\hat{\theta}_{ch,dB}$, $\hat{\alpha}_{ch,dB}$, $\hat{\beta}_{ch}$, and $\hat{\sigma}_{ch,dB}$ are the estimated channel parameters, $H_q = [1 - 10\log_{10}(\|q - q_b\|)]$, $H_{Q_m} = [H_{q_1}^T, \dots, H_{q_m}^T]^T$, $\Psi(q) = [\hat{\alpha}_{ch,dB}^2 \exp(-\|q - q_1\|/\hat{\beta}_{ch}), \dots, \hat{\alpha}_{ch,dB}^2 \exp(-\|q - q_m\|/\hat{\beta}_{ch})]^T$, and $\Phi = \Omega + \hat{\sigma}_{ch,dB}^2 I_m$ with $[\Omega]_{i,j} = \hat{\alpha}_{ch,dB}^2 \exp(-\|q_i - q_j\|/\hat{\beta}_{ch})$, $\forall i, j \in \{1, \dots, m\}$ and I_m denoting the identity matrix.

This formulation allows the robot to predict the channel quality at any unvisited locations, based on a small number of prior channel measurements in the same environment.¹ See [1] for more details and the performance of this channel predictor in different real environments.

Based on this channel prediction, the expected required transmission power at location q is given by $\mathbb{E}[\Gamma_T(q)] = (2^r - 1)\mathbb{E}[1/\Upsilon(q)]/Z$, where $\Upsilon(q)$ is the predicted CNR at location q (in non-dB domain), $\mathbb{E}[1/\Upsilon(q)]$ can be evaluated given the log-normal distribution of $\Upsilon(q)$ (or Gaussian in dB), and $Z = -1.5/\ln(5p_{\text{BER}})$. Given a transmission time duration, the robot can then predict the communication energy cost for transmitting

¹It should be noted that the robot needs to predict the uplink channel for the purpose of its sensing-path-communication co-optimization, as we shall see. Thus, the small number of prior channel measurements can be collected by the remote station and transmitted back to the robot which will be in charge of channel prediction over the space. Alternatively, if Time Division Duplex is used, the robot can directly use a small number of downlink channel samples in order to predict the channel elsewhere.

from any unvisited location to the remote station. This prediction is important for the robot's path planning, as we shall see.

2.2 Robot Motion Energy Model [2]

Based on experimental studies, a mobile robot's motion power can be modeled by a linear function of its speed for a number of platforms [2]: $\Gamma_M = \kappa_1 u + \kappa_2$ when $0 < u \leq u_{\max}$, and $\Gamma_M = 0$ when $u = 0$, where u and u_{\max} are the robot's speed and maximum speed, respectively, and κ_1 and κ_2 are positive constants determined by the robot's motor, mechanical transmission system, and external load. Suppose that the robot travels at a constant speed u_{const} . The motion energy cost for a travel distance of l is then given by $\mathcal{E}_m = (\kappa_1 + \kappa_2/u_{\text{const}})l$. This is the model we adopt in this dissertation.

Chapter 3

Prediction of Human Visual Performance

In this chapter, we are interested in robotic surveillance tasks that involve visual perception, where the robot has a limited access to a remote operator to ask for help with its task. Since human visual performance is not perfect and varies depending on the quality of the visual input, in order for the robot to optimally query the human operator and plan its field actions, it needs to predict human visual performance for a given visual input. We then propose a machine learning-based approach that allows the robot to probabilistically predict human visual performance. First, we look at a simple setting where the main source of visual uncertainty is the image noise, which brings interesting insights about human visual performance, as well as about the optimum collaboration with the human. Then, we consider a general case with any type of visual uncertainty and utilize a Deep Convolutional Neural Network (DCNN) to probabilistically predict human visual performance, without the need for hand-crafting any features. This DCNN-based predictor enables the robot to select the right questions, only querying the operator with the images for which humans have a high chance of success. The robot then utilizes this

Parts of this chapter have appeared in our papers [7], ©[2015] IEEE, and [8].

prediction information to optimize its field decisions, in terms of when to rely on itself, when to ask for human help, and when to further sense the field. Our several real-world robotic experiments demonstrate that by using our proposed human performance predictor, the overall visual recognition performance significantly outperforms a benchmark that assumes perfect human performance.

The rest of this chapter is organized as follows. In Sec. 3.1, we look at the case of predicting human performance based on image noise and analyze the optimum robot behavior. In Sec. 3.2, we present our proposed DCNN-based approach for predicting human visual performance in a general setting. In Sec 3.3, we then show how we can incorporate this prediction capability into the robot’s field decision-making for optimum collaboration with the human operator. Finally, we validate our work with several experimental results. We describe the robot vision model in Sec. 3.4 and then present several results from real-world robotic experiments to demonstrate the effectiveness of our human prediction approach for human-robot collaborations in Sec. 3.5.

3.1 Assessing Human Visual Performance on Noisy Images

Consider the case where the robot has discovered a target via visual sensing (e.g., taking an image) and needs to classify it based on a given set of target possibilities. For example, Fig. 3.1 shows four possible targets shown to the robot. The robot’s sensing in the field is in general subject to noise, low resolution, occlusion, and other uncertainties, which will degrade its classification accuracy. Fig. 3.2 shows a sample case where an image is corrupted by an additive Gaussian noise with a variance of 3. Based on Amazon Mechanical Turk (MTurk) [30] studies, 74.4% of humans can still classify this

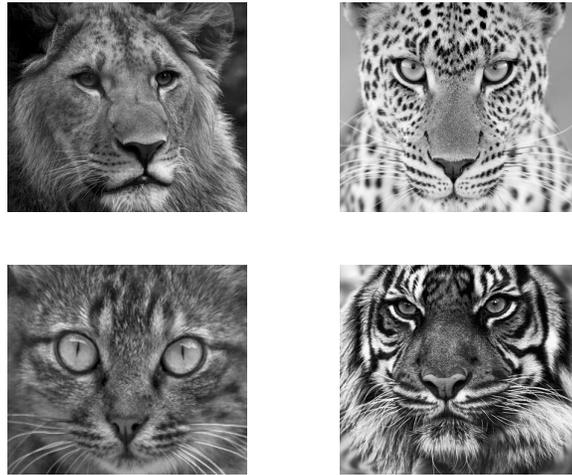


Figure 3.1: Gray-scale images of lion, leopard, cat, and tiger used in our study.

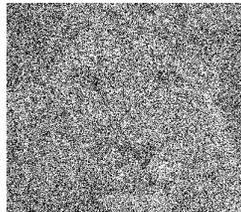


Figure 3.2: A sample corrupted image (lion) with the noise variance of 3.

lion correctly (see Fig. 3.3). If the robot could accurately model all the uncertainties and use the optimum classifier accordingly, it would outperform humans. However, this is impossible due to the complexity of a real-life visual task. This is why the robot can benefit tremendously from collaborating with the human by properly taking advantage of human visual abilities. In this section, we study a case where the main source of uncertainty in the robot's visual sensing is a Gaussian-like noise

3.1.1 User Studies to Assess Human Performance

Consider the case where the robot takes an image in the field, which is corrupted by an additive Gaussian noise with a known variance but an unknown mean, and then undergoes a pixel value clamping process unknown to the robot. The original image is chosen from

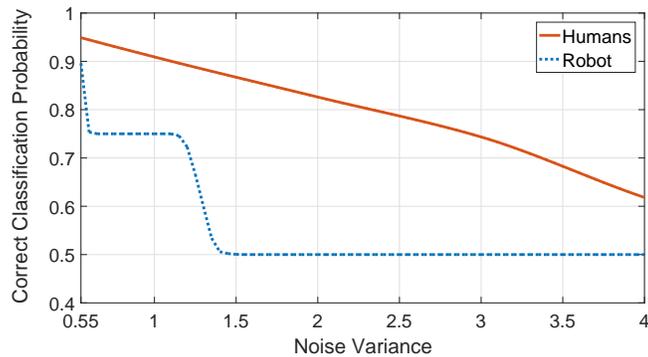


Figure 3.3: Performance of human and robot in classifying the noisy targets of Fig. 3.1.

the four images in the Fig. 3.1. Then, a total of 160 noisy images were generated with different noise levels. We then conducted extensive user studies on Amazon MTurk, where, for each noisy image, several human users were asked to classify the target in the image to one of the four possibilities in Fig. 3.1. Overall, 8,000 human responses were collected, based on which the probability of human correct classification as a function of the noise level was obtained, as shown in Fig. 3.3. The figure further compares the human performance with that of the robot, and confirms that humans can achieve a much higher correct classification probability. For instance, for Fig. 3.2, humans have an estimated correct classification probability of 0.744, which is considerably higher than that of the robot (0.5). However, as the figure also shows, human performance is not perfect, and can degrade significantly as the noise level increases.

3.1.2 Optimizing Human-Robot Collaboration in a Target Classification Task

In this section, we consider a simple scenario of human-robot collaboration in a target classification task, in order to bring a foundational understanding to the characteristics of the optimal collaboration. Suppose that the robot has an initial sensing (in the form of an acquired image) of N given sites. The robot is given a total motion energy budget

as well as a total number of questions to ask the human (dictated by the frequency of the availability of communication with the human). For each site, the robot has three choices: 1) rely on its own classification (based on the initial sensing), 2) use a question and present the image of the site to the human, or 3) spend motion energy to go to the site and sense better.

Problem Setup

Consider the case where we have a total number of N sites. Each site contains one of N_c a priori known targets (see Fig. 3.1 for an example with $N_c = 4$ targets). For each site, the robot acquires an image which is then corrupted by an additive Gaussian noise with an unknown mean but a known variance and pixel values of the image are then truncated. The robot uses the minimum distance detector [7], which would have been ideal for the case of a zero-mean additive Gaussian noise with a known variance. The robot is allowed to query the human M times and has a total motion energy budget of \mathcal{E}_{\max} . The probabilities of correct target classification of site i , for $i \in \{1, \dots, N\}$, are denoted by $p_{r,i}$ and $p_{h,i}$ for the robot and the human, respectively. These probabilities are obtained from Fig. 3.3, based on the variance assessed by the robot. Let \mathcal{E}_i denote the motion energy cost to visit site i , which can be evaluated by the robot. If the robot chooses to visit a site, the probability of correct classification increases to a high value of \tilde{p} . The objective of the robot is then to decide which sites to present to the human, which sites to visit and which sites to rely on its own classification based on its initial sensing, in order to maximize the overall average probability of correct classification under resource constraints. Let P_{correct} denote the average probability of correct classification of a site.

We have

$$\begin{aligned}
P_{\text{correct}} &= \frac{1}{N} \left(\sum_{i=1}^N \gamma_{h,i} p_{h,i} + \sum_{i=1}^N \gamma_{s,i} \tilde{p} + \sum_{k=1}^N (1 - \gamma_{h,i})(1 - \gamma_{s,i}) p_{r,i} \right), \\
&= \frac{1}{N} \left(\sum_{i=1}^N \gamma_{h,i} (p_{h,i} - p_{r,i}) + \sum_{i=1}^N \gamma_{s,i} (\tilde{p} - p_{r,i}) + \sum_{i=1}^N p_{r,i} \right),
\end{aligned}$$

where $\gamma_{h,i}$ is 1 if the robot seeks human's help for site i and is 0 otherwise. Similarly, $\gamma_{s,i} = 1$ indicates that the robot will physically visit site i and $\gamma_{s,i} = 0$ denotes otherwise.

We then have the following optimization problem:

$$\begin{aligned}
&\max_{\gamma_h, \gamma_s} \gamma_h^T (p_h - p_r) + \gamma_s^T (\tilde{p} \mathbf{1} - p_r) \\
&\text{s.t.} \quad \gamma_s^T \mathcal{E} \leq \mathcal{E}_{\max}, \quad \mathbf{1}^T \gamma_h \leq M, \\
&\quad \gamma_h, \gamma_s, \gamma_h + \gamma_s \in \{0, 1\}^N,
\end{aligned} \tag{3.1}$$

where $p_h = [p_{h,1}, \dots, p_{h,N}]^T$, $p_r = [p_{r,1}, \dots, p_{r,N}]^T$, $\gamma_h = [\gamma_{h,1}, \dots, \gamma_{h,N}]^T$, $\gamma_s = [\gamma_{s,1}, \dots, \gamma_{s,N}]^T$, $\mathcal{E} = [\mathcal{E}_1, \dots, \mathcal{E}_N]^T$, and $\mathbf{1}$ is the vector with all 1s.

It can be seen that $(p_{h,i} - p_{r,i})$ and $(\tilde{p} - p_{r,i})$ are important parameters as they represent the performance gains by asking the human and visiting site i , respectively. Note that we do not allow the robot to both query the human and make a visit for the same site. This is because we already assume a high probability of correct classification when the robot visits a site. Thus, allowing the robot to both visit and ask about the same site will be a waste of resources in this case.

Properties of Optimal Collaboration

In this section, we discuss some properties of the optimization problem (3.1). We first consider a special case where the exact solution can be derived and then show a few properties of the general optimum solution.

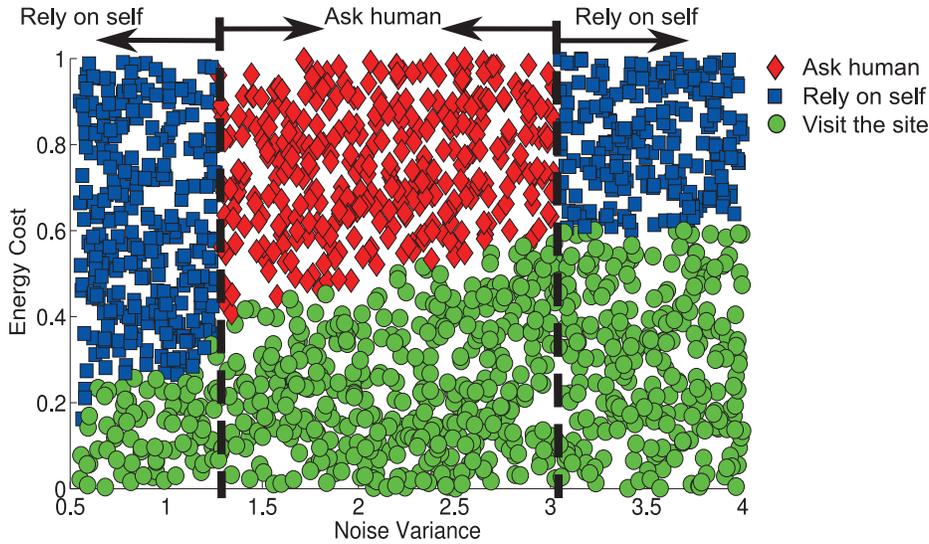


Figure 3.4: An example of the optimum decisions with 2,000 sites, 500 questions and an energy budget of 25% of the total energy needed to visit all the sites.

If $\mathcal{E}_{\max} = 0$, problem (3.1) reduces to the case where the robot needs to decide between asking the human and relying on its initial classification. It can then be easily confirmed that $\gamma_{h,i}$ has to be one for the M sites with the largest $p_{h,i} - p_{r,i}$ in this case.

In general, optimization problem (3.1) is an integer program, which makes theoretical analysis challenging. In order to better understand the optimum solution of problem (3.1), Fig. 3.4 shows an example of the optimum decisions for the case of $N = 2,000$ sites, with $M = 500$ allowed questions and an energy budget equal to 25% of the total energy needed to visit all the sites. The optimum decision for each site is marked based on solving problem (3.1). Interesting behavior emerges as can be seen. For instance, we can observe that there are separations between different decisions. The clearest patterns are two transition points that mark when the robot asks human for help, as shown with the dashed vertical lines in Fig. 3.4. Basically, the figure suggests that the robot should not query the human if the variance is smaller than a threshold or bigger than another threshold, independent of the motion cost of a site. This makes sense as the robot itself

will perform well for low variances and human does not perform well for high variances, suggesting an optimum query range. Furthermore, it shows that the robot only visits the sites where the energy cost is not too high and relies more on itself for the sites with both high variance and energy cost.

3.2 Predicting Human Visual Performance Using Deep Learning

In this part, we develop a deep learning-based algorithm to predict human visual performance under general visual uncertainties, for a task that involves finding a human in a site. More specifically, the robot is given the task of finding the person in a number of sites based on its onboard camera input. In order to properly ask for help from the remote operator for the sites for which it cannot find the person, the robot needs to predict human visual performance. In this part, we show how to develop a DNN pipeline in order to predict human visual performance in this task, given any general visual sensing uncertainty. More specifically, given a visual input in the form of an image with a person in it, the robot wants to predict the probability that humans can find the person in the image. In order to equip the robot with this capability, we collect several images with different levels of difficulty to train a Deep Convolutional Neural Network (DCNN)-based human performance predictor, as we will show in this section. This trained DCNN-based predictor then enables the robot to ask the right questions, only querying the operator with the sensory inputs for which humans have a high chance of success, as well as optimize its field decisions in terms of relying on itself, asking for human help, and further sensing, which we will discuss in details in Sec. 3.3.

In this section, we describe the details of the training process, the machine learning

algorithm, as well as the prediction performance.

3.2.1 Training Dataset

We have built a dataset of 3,000 total images, with each image containing a human.¹ We have included images with different degrees of visual difficulty for proper training. Sample challenging cases include images in cluttered environments, images where the human is far away, and dark images. The images are mainly selected from the following sources: NOAA Natural Hazards Image Database [108], the SUN dataset [109], and the PASCAL VOC dataset [110]. All the images are resized to 256×256 . For each acquired image, we have manually darkened it to create a new data point that is also dark.

Human Data Collection

For each training image, we need to evaluate human visual performance. We do so empirically by using MTurk. For each image, 50 MTurk workers are asked if they could see a human in the image. The responses are then averaged for each image to empirically assess the probability that a person can accomplish the given visual task.

Statistical Characteristics of the Dataset

Ideally, our dataset should be balanced in the number of difficult and easy images to avoid biasing the learning process. However, we find more easy images in online datasets than hard ones. For instance, based on the collected data from MTurk, the probability of task accomplishment is above 0.95 for about 70% of the images and below 0.9 for

¹Ideally each image should contain only one human to be consistent with the defined task. However, due to the difficulty of finding images online that capture challenging scenarios, a small percentage of the images contain more than one person in the image. The MTurk users are instructed to answer “yes” if they could spot a person in the image. In such cases, the task can be thought of as finding the most obvious person in the image.

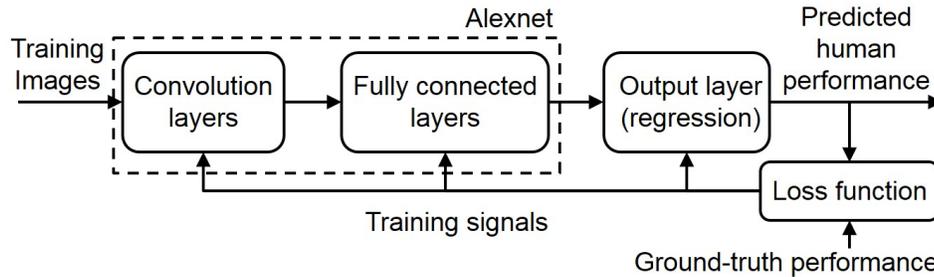


Figure 3.5: Flow diagram of training a convolutional neural network to predict human performance probabilistically.

about 20% of the images. To avoid biasing the prediction, we utilize the commonly used technique of oversampling [111] in machine learning to create a more balanced dataset. As our extensive experimental results of Section 3.5 indicate, the number of hard images in the initial set is still sufficient as our trained predictor can well differentiate between the images that are easy and difficult for humans.

3.2.2 Using DCNN to Predict Human Performance

We train a convolutional neural network to predict human performance. The high-level structure of the network is shown in Fig. 3.5. Our proposed network architecture is a modified version of AlexNet [48], which is one of the state-of-the-art DCNNs. The original AlexNet consists of 5 convolutional layers and 2 fully connected layers, followed by an output layer. AlexNet is originally designed for object classification. In this chapter, however, we are interested in a different task of probabilistically predicting human performance in finding a person in an image, which is a regression task. We thus replace the output layer of AlexNet with a regression layer, which then outputs the human performance based on the output of the 7th layer of AlexNet.

By using the dataset of Section 3.2.1 as input images and the corresponding MTurk responses as ground-truth labels, we train the resulting network as follows. The parameters of the first 7 layers are initialized with the weights of the corresponding layers of an

Image categories	Normalized MSE
All	0.0199
Non-dark	0.0067
Dark	0.0330
Hard	0.0479
Hard Non-dark	0.0493
Hard Dark	0.0474

Table 3.1: Performance of our human predictor – NMSEs over different subsets of images in the validation set.

AlexNet trained for human detection (classification from a set of {Human, No Human}). The parameters of the output layer are initialized randomly. Mean Squared Error (MSE) is used as the loss function that computes the update for network weights. The underlying motivation for this design is as follows. The parameters of the first 7 layers of the AlexNet trained for human detection capture features useful for detecting human presence, which are also informative for predicting human performance in finding a person in the image. But since our task is different, all the weights need to be updated during the training based on the input from human performance. The training is performed for 50,000 iterations with an initial base learning rate of 10^{-5} , which is reduced by a factor of 10 after every 10,000 iterations. Stochastic gradient descent is used to update the network weights and the training batch size is 128. We employ the machine learning library Caffe [112] to train and test our models.

3.2.3 Evaluation of the Proposed Human Performance Predictor

We next evaluate the performance of our trained human predictor over the validation set. The validation set is randomly selected out of the original image pool and thus has a similar ratio of hard to easy images as the training pool. We first look at the Normalized Mean Squared Error (NMSE) to evaluate the prediction quality, which is calculated as

follows: $\frac{1}{N_v} \sum_{i=1}^{N_v} (p_{h,i,\text{val}} - \hat{p}_{h,i,\text{val}})^2 / p_{h,i,\text{val}}^2$, where $p_{h,i,\text{val}}$ and $\hat{p}_{h,i,\text{val}}$ are the ground-truth and predicted human performance of the i^{th} validation image and N_v is the number of image in the validation set.² Table 3.1 shows a summary of NMSE over different subsets of the validation set. The NMSE over all validation images is 0.0199. The NMSE over all non-dark and dark³ validation images are 0.0067 and 0.033 respectively. It is also of great importance to our robotic task to be able to predict images hard for humans. Define hard images such that the empirical probability of a person finding the human in the image, i.e., the ground-truth probability, is less than 0.9. The NMSEs over all such hard images, hard non-dark images and hard dark images are also summarized in Table 3.1. As can be seen, the NMSE values are fairly small, indicating a good training performance. Fig. 3.6 shows four sample images with the true and predicted human performance annotated (bottom-right) on each image. As can be seen, our predictor can predict the performance well for these images. Fig. 3.7 further shows the empirical Cumulative Distribution Function (CDF) of NMSE over the validation set. It can be seen that the NMSEs of most images (more than 90%) are upper bounded by 0.05. We note that the NMSE of the validation set and the training set is 0.0199 and 0.0112 respectively, indicating that the network did not over-fit the training data.

In the literature, there are two existing papers that have looked into the problem of predicting the probability with which a person is able to see a human in an image. The reported prediction Mean Absolute Error (MAE) is at least 0.22 in [5], while the prediction Mean Squared Error (MSE) is 0.04 in [4]. The comparison between our predictor and those reported in [4, 5] is summarized in Table 3.2. As can be seen, our approach can achieve a much better performance. This is mainly because our approach

²Although NMSE values are higher than the corresponding MSE ones since we are predicting a positive value bounded by 1, we find NMSE a more truthful metric of the performance as it is normalized by the true value.

³Non-dark images refer to those directly taken from the aforementioned datasets and dark images refer to those that are manually darkened.



Figure 3.6: Comparison of the true and predicted probability of successful human performance for (top-left) an easy non-dark image, (top-right) a hard non-dark image, (bottom-left) an easy dark image and (bottom-right) a hard dark image. Readers are referred to the color pdf for better viewing.

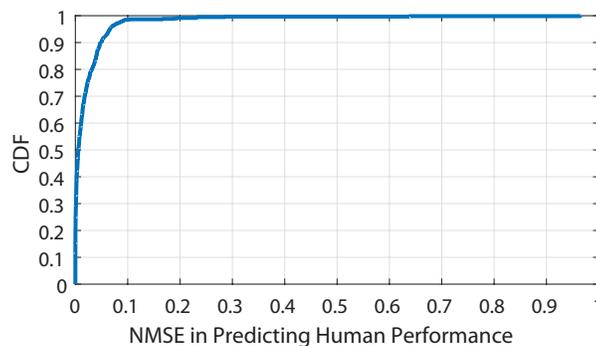


Figure 3.7: The CDF of NMSEs of validation images.

is fundamentally different from the past work in that [4, 5] choose a few hand-crafted image features for prediction. In general, however, there exist many underlying features that contribute to what makes an image difficult or easy for human visual system, which are challenging to hand-craft. Thus, in our approach, the important features of an image are automatically captured by the DCNN in Fig. 3.5, and utilized for performance prediction.

Human Predictors	MSE	MAE
Our Predictor	0.0067	0.0459
Predictor of Engel et al.	0.04	N/A
Predictor of Wakayama et al.	N/A	0.22

Table 3.2: Comparison of the performance of our predictor with two existing predictors. N/A indicates that the evaluation is not available. The performance of our predictor is evaluated over all non-dark validation images, which best match the characteristics of the images used in [4, 5].

Remark 1: Table 3.2 shows what [4, 5] reported for their prediction performance. Although we evaluate the performance of our predictor over our non-dark validation images, which best resemble the images used in [4, 5], we note that our dataset is not the same as that of [4] or [5]. The performances of the three predictors are thus not fully comparable since they are evaluated over different datasets.

Remark 2: As can be seen, the NMSE values in Table 3.1 are slightly larger over the set of hard images. While we expect hard images to be more challenging for the predictor, this can also be partly due to the dataset imbalance (discussed in Section 3.2.1). It can, however, be seen that the NMSE values are still very small, indicating a good training performance. Building a more balanced dataset by including more hard images, as part of future work, can improve the prediction performance in more challenging settings.

3.3 Robot Decision Optimization

In this section, we formulate the human-robot collaborative perception problem as a constrained optimization problem and discuss the optimum decisions such that robot’s motion energy usage is minimized while task performance is guaranteed above a certain level. We then propose how to equip the robot with the human performance predictor in order to successfully execute the optimum decisions. In the rest of the chapter, we focus on surveillance tasks for finding humans based on imagery inputs. We note that

the proposed methodology is applicable to any visual perception task with any sensory input.

3.3.1 Problem Formulation

Consider a case where there is a total of N sites with a human at each site. A field robot is tasked with finding the human at each site.⁴ The robot has limited access to a human operator to ask for help with the task, in the form of M maximum questions. The robot can also spend motion energy and time to move closer to a site for better sensing. The robot's goal is to successfully perform the task while minimizing its total energy usage (or equivalently operation time).

During the operation, the robot first performs initial sensing of the sites by taking a picture of each site. Based on these sensory inputs, it then estimates its own probability of task accomplishment, which is denoted by $p_{r,i}$ for site i , for $i \in \{1, \dots, N\}$. If the estimated probability is high enough for a site, then the robot can rely on itself. If not, it has to decide if it should ask for help from a remote operator for this site or if it should move to the site for further sensing. In order to properly make this decision, it needs to assess the chance that the operator can perform the task successfully. Let $p_{h,i}$ denote the probability that humans can successfully perform the task for site i , for $i \in \{1, \dots, N\}$. We then have the following optimization problem:

$$\begin{aligned}
 & \min_{\gamma_h, \gamma_r, \gamma_s} \mathcal{E}^T \gamma_s \\
 & \text{s.t.} \quad \gamma_h \circ p_h + \gamma_r \circ p_r + \gamma_s \succeq p_{\text{Th}} \mathbf{1}, \\
 & \quad \mathbf{1}^T \gamma_h \leq M, \quad \gamma_h + \gamma_s + \gamma_r = \mathbf{1}, \quad \gamma_h, \gamma_s, \gamma_r \in \{0, 1\}^N,
 \end{aligned} \tag{3.2}$$

⁴We emphasize that the considered task in the rest of the chapter is to find the human at each site, given there is a human at each site. In other words, the robot or human operators/MTurk users know there should be a human at each site and the task is to find the human at each site.

Algorithm 1: Decision Making Algorithm

Initialization: $\gamma_{r,i} = 0, \gamma_{h,i} = 0, \gamma_{s,i} = 0, \forall i \in \{1, \dots, N\}$.

Step 1: $\forall i \in \{j : p_{r,j} \geq p_{\text{Th}}, j \in \{1, \dots, N\}\}$, set $\gamma_{r,i} = 1$;

Step 2: $\forall i \in \{j : \gamma_{r,j} = 0, p_{h,j} \geq p_{\text{Th}}, j \in \{1, \dots, N\}\}$, set $\gamma_{h,i}$ with the M largest \mathcal{E}_i to 1;

Step 3: $\forall i \in \{j : \gamma_{r,j} = \gamma_{h,j} = 0, j \in \{1, \dots, N\}\}$, set $\gamma_{s,i} = 1$.

where $\mathcal{E} = [\mathcal{E}_1, \dots, \mathcal{E}_N]^T$ is the motion energy cost vector to visit the sites, p_{Th} is the minimum acceptable probability of successful task accomplishment, $p_h = [p_{h,1}, \dots, p_{h,N}]^T$, $p_r = [p_{r,1}, \dots, p_{r,N}]^T$, $\gamma_h = [\gamma_{h,1}, \dots, \gamma_{h,N}]^T$, $\gamma_s = [\gamma_{s,1}, \dots, \gamma_{s,N}]^T$ and $\gamma_r = [\gamma_{r,1}, \dots, \gamma_{r,N}]^T$. Moreover, $\gamma_{h,i} = 1$ if the robot seeks human's help for site i and $\gamma_{h,i} = 0$ otherwise. $\gamma_{s,i} = 1$ if the robot visits site i and $\gamma_{s,i} = 0$ otherwise. Furthermore, $\gamma_{r,i} = 1$ if the robot relies on itself for site i and $\gamma_{r,i} = 0$ otherwise. $\mathbf{1}$ is the vector of all 1s, \circ denotes the Hadamard product and \preceq indicates that the inequality is component-wise.

In order to mathematically characterize the optimum decisions, we assume that if the robot moves to a site for further sensing, its probability of successful task accomplishment is 1 in problem (3.2).⁵ The optimum solution of Algorithm 1 can then be easily confirmed for problem (3.2).

3.3.2 Proposed Decision Optimization

Algorithm 1 lays out the decision optimization of the robot during the operation. If the robot has a high confidence in finding the person at a site ($p_r \geq p_{\text{Th}}$), then it can simply rely on itself, eliminating the need for further sensing or asking for help. In Section 3.4, we discuss how robot's vision algorithm provides it with p_r so it can assess its own performance.

For sites that $p_r < p_{\text{Th}}$, the robot has to decide on which ones to visit and which

⁵In our experiments, we take the true probability of task accomplishment into account when a site is visited.

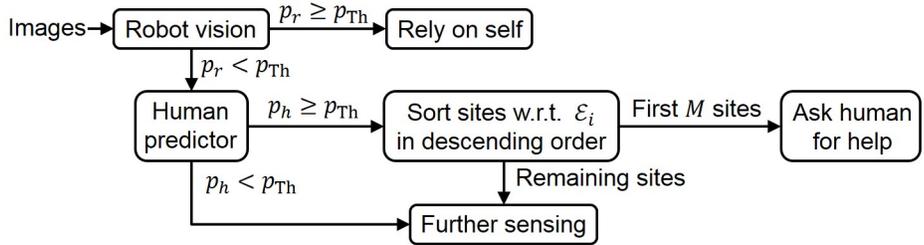


Figure 3.8: Flow diagram of robot’s decision making process.

ones to ask the operator. In order to make this decision, the robot needs to predict the probability that the operator can accomplish the given perception task (p_h). This probability is estimated by our trained DCNN human visual performance predictor, as extensively discussed in Sec. 3.2. Based on Algorithm 1, the robot then evaluates the images of these sites with its human predictor. Let N_{hard} denote the number of sites that the robot cannot rely on itself. Out of these sites, let N_{human} denote the number of sites for which the robot predicts that the human can accomplish the task ($p_h \geq p_{\text{Th}}$). If $M \geq N_{\text{human}}$, then the robot will pass all those sites to the operator. If $M < N_{\text{human}}$, then the robot will choose the M sites that cost the most energy to visit and pass them to the operator. It will then visit the remaining sites. Fig. 3.8 summarizes the robot’s optimum decision-making.

3.3.3 Variability Among Human Operators and p_{Th}

In general, we should choose p_{Th} high such that the robot only asks the human about those images for which almost all the probed people have correctly done the task. This is in particular important as we consider cases where the human operator can only look at the image at a glance and does not have time to investigate the details due to work overload. Thus, the robot wants to only ask for help if it is fairly confident that the human can be of help.

Choosing a high threshold also makes the prediction more immune to the variabil-

ity among operators. As expected, there will be variability among different humans in accomplishing a visual task. Furthermore, performance of the same person can vary depending on factors such as fatigue and attention overload. The variability of performance, however, is much less for easy images, as compared to hard images. This makes sense as humans can typically perform an easy task even under fatigue and stress while performing harder tasks requires more focus and energy. We thus choose $p_{\text{Th}} = 0.9$ in the next section on performance evaluation.

3.4 Robot Vision

For the given visual task of finding a human in an image, the robot uses the state-of-the-art AlexNet [48], which also provides it with a confidence metric to assess its probability of task accomplishment, denoted as p_r . We train AlexNet to classify images into two classes of {Human, Non-Human}, i.e., images that contain at least one human and images that do not contain any. We fine-tune the AlexNet on our dataset (MS COCO [76] plus manually-darkened images) with 230,000 training images from a model pre-trained on the Imagenet data [113]. The accuracy of our model over the validation set (80,000 images) is 0.85. The network gives its confidence in terms of a probability for each class. We then take $p_r = \text{prob}\{\text{Human}\}$.

3.5 Performance Evaluation

In this section, we start by evaluating the performance of the proposed approach with 4 experiments on our campus. Additionally, we evaluate the performance at a larger scale by simulating a case with 15 sites, based on real sensing data from our campus and real human performance using MTurk. We further compare our approach

with the best possible without human prediction, to which we refer as “benchmark”. The benchmark optimizes the decisions without knowledge of human performance, as is summarized below:

Initialization: $\gamma_{h,i} = 0$, $\gamma_{s,i} = 0$ and $\gamma_{r,i} = 0$, $\forall i \in \{1, \dots, N\}$;

Step 1: $\forall i \in \{j : p_{r,j} \geq p_{Th}, j \in \{1, \dots, N\}\}$, set $\gamma_{r,i} = 1$;

Step 2: $\forall i \in \{j : \gamma_{r,j} = 0, j \in \{1, \dots, N\}\}$, set $\gamma_{h,i}$ with the M highest $p_{r,i}$ to 1;

Step 3: $\forall i \in \{j : \gamma_{r,j} = \gamma_{h,j} = 0, j \in \{1, \dots, N\}\}$, set $\gamma_{s,i} = 1$.

In summary, the robot orders the sites based on its own confidence (p_r) from highest to lowest. If the robot’s confidence is high enough, it will rely on itself. It then selects the next M sites to query human operators, and visits the remaining sites. We note that in several literature on human-robot collaboration, the human is assumed perfect. We can also compare our approach to the case of perfect human. In this case, the robot should choose M most expensive sites to query the operator from the sites that are difficult for itself. However, this approach does worse than the aforementioned benchmark. We thus compare our approach with the described benchmark in this section. When comparing to any other approach, we note that the sites that the robot relies on itself will naturally be the same, as expected, since this chapter is about how to optimize the decisions when the robot cannot rely on itself.

3.5.1 Robotic Experiments

We perform a number of robotic experiments at different locations on our campus to validate the proposed approach. In each experiment, the robot starts at the center of a crossroad. Each crossroad direction is a site of interest, and there is a person at each site. The robot is tasked with finding the person in each direction based on camera inputs, and is given maximum of $M = 1$ question to ask remote operators for help with

finding the person in that experiment. The robot can choose to move to a site for further sensing. The goal of the robot is to find the person in each direction with a very high probability, while minimizing its total energy usage.

The robot does an initial sensing of each direction by rotating, facing each direction, and taking a color picture. The robot then inputs the pictures to its onboard human predictor and decision making algorithm (problem (3.2)). Based on the resulting optimum decisions, the robot may select a site to ask for operator’s help and a number of sites for further sensing according to the strategy described in Section 3.3.2. The robot is a Pioneer 3-AT ground vehicle [114], equipped with a webcam for sensing and a laptop for processing. Camera images are resized to 256×256 to be compatible with AlexNet. For all the images, the readers are referred to the color pdf.

Campus Robotic Experiment 1

In this experiment, the robot starts at crossroad 1 with three sites in three different directions. Fig. 3.9 shows the images that the robot takes during its initial sensing. Table 3.3 shows robot’s performance (p_r) as well as predicted human performance (\hat{p}_h), as compared to the required threshold (0.9) for the three sites, with “1” indicating that the threshold is satisfied and “0” denoting otherwise.⁶ As can be seen, the robot can confidently rely on itself for Sites 2 and 3. However, it has a low confidence for Site 1, which is also hard for humans, as can be seen. Without our predictor, however, the robot has no methodical way of making the right decision for this site. The table shows robot’s decisions for both our approach and the benchmark. As can be seen, our human predictor accurately predicts that Site 1 is too difficult for humans and thus does not send this image to the operators. Instead, it chooses to move to this site to take the 4th image, which is now easy for itself. The benchmark, on the other hand, would inquire the

⁶The threshold $p_{Th} = 0.9$ is used throughout this section.



Figure 3.9: Initial sensing images of 3 sites at crossroad 1 and the image taken after moving to Site 1 for further sensing based on our approach. Readers are referred to the color pdf for all the experimental results. The visual difficulty of the real-size images is pretty consistent with that of the smaller versions when viewed in color.

Site	$p_r \geq p_{Th}$	$\hat{p}_h \geq p_{Th}$	Our Method	Benchmark
1	0	0	Visit	Ask
2	1	1	Self	Self
3	1	1	Self	Self
Ave. Prob. Success			0.96	0.69
Ave. Prob. Human Success			N/A	0.07

Table 3.3: Performance summary at campus crossroad 1.

operator on the first image, for which MTurkers had 0.07 chance of spotting the human. The table also shows the overall average probability of task accomplishment, which is 0.69 for the benchmark and 0.96 for our approach (39% higher than benchmark). Average probability of success is averaged over all the sites, including the ones that the robot relies on itself.

Campus Robotic Experiment 2

In this experiment, the robot starts at crossroad 2 with three sites in three different directions. Fig. 3.10 shows the images taken by the robot during its initial sensing. Table 3.4 shows robot’s performance and predicted human performance. As can be seen, the robot can confidently rely on itself for Site 3. However, it has a low confidence for Sites 1 and 2 in this case. Site 1, however, is also hard for humans, while Site 2 is easy for



Figure 3.10: Initial sensing images of 3 sites at crossroad 2 and the image taken after moving to Site 1 for further sensing based on our approach.

Site	$p_r \geq p_{Th}$	$\hat{p}_h \geq p_{Th}$	Our Method	Benchmark
1	0	0	Visit	Ask
2	0	1	Ask	Visit
3	1	1	Self	Self
Ave. Prob. Success			1	0.71
Ave. Prob. Human Success			1	0.27

Table 3.4: Performance summary at campus crossroad 2.

humans, as can be seen.⁷ The table shows robot’s decisions for both our approach and the benchmark. Our human predictor accurately predicts that Site 2 is easy for humans while Site 1 is hard. The robot then chooses to move to Site 1 and send image of Site 2 to operators for help, while relying on itself for Site 3.

The benchmark instead queries the operator with Site 1 and moves to Site 2 for further sensing, which results in wasting one question, not spotting the person in Site 1, and moving to the wrong site for further sensing. This is due to the fact that p_r of Site 1 is higher than Site 2. In other words, robot’s vision algorithm cannot properly predict human’s performance. As a result, the average probability of human success is 0.27 in the benchmark case while it is 1 in our case (3.7 times higher). The overall average probability of success is 0.71 for the benchmark and 1 for our approach (41% higher than benchmark) in this case.

⁷It is easy to spot the person in this image in the color pdf.



Figure 3.11: Initial sensing images of 4 sites at crossroad 3 and the image taken after moving to Site 1 for further sensing based on our approach.

Site	$p_r \geq p_{Th}$	$\hat{p}_h \geq p_{Th}$	Our Method	Benchmark
1	0	0	Visit	Ask
2	1	1	Self	Self
3	1	1	Self	Self
4	0	1	Ask	Visit
Ave. Prob. Success			1	0.77
Ave. Prob. Human Success			1	0.12

Table 3.5: Performance summary at campus crossroad 3.

Campus Robotic Experiment 3

In this experiment, the robot starts at crossroad 3 with four sites in four different directions. Fig. 3.11 shows the initial images. As can be seen, it is hard to spot the person in Site 1, while the person in the three other sites can be easily detected. Table 3.5 shows robot’s performance and predicted human performance. The predictor flags the first site as hard for humans. Then, our approach results in the robot asking about Site 4, visiting Site 1, and relying on itself for the remaining two sites. The benchmark moves to Site 4 instead and inquires operators on Site 1. The average probability of human success is 0.12 in the benchmark case while it is 1 (8.3 times higher) in our case. The overall average task accomplishment probability is 1 for our approach, 30% higher than the benchmark (0.77) in this case.



Figure 3.12: Initial sensing images of 3 sites at crossroad 4 and the image taken after moving to Site 1 for further sensing based on our approach.

Campus Robotic Experiment 4

Next, we show a case where the benchmark gets lucky and has the same decisions as our approach. In this experiment, the robot starts at crossroad 4 with three sites, as shown in Fig. 3.12. Robot’s vision fails for Sites 1 and 3. As can be seen, it is hard to see the person in the first image and easy to spot the person in the other two. Our approach correctly predicts this and visits Site 1 while asking operators about Site 3. The benchmark coincides with our approach in this case. However, in general, it will be hard for the robot to ask for help without a proper human predictor.

Overall, our experimental campus results confirmed that the human predictor can properly help the robot identify which images are hard/easy for humans, allowing for the optimization of further sensing and human query. Unless all the sites are easy for the robot or for the human, the robot cannot methodically optimize its decisions without properly predicting human performance. In practice, there will be several hard cases where the robot cannot rely on itself and the sensory input is still too hard for humans, as we have shown. The proposed predictor and decision optimization approach can then be a valuable tool for the robot to achieve its best performance with minimum resources.

	Sites 1-5	Sites 6-10	Sites 11-15
Robot Perf.	$p_r < p_{Th}$	$p_r < p_{Th}$	$p_r \geq p_{Th}$
Human Perf.	$p_h < p_{Th}$	$p_h \geq p_{Th}$	$p_h \geq p_{Th}$
Predicted Human Perf.	$\hat{p}_h < p_{Th}$	$\hat{p}_h \geq p_{Th}$	$\hat{p}_h \geq p_{Th}$

Table 3.6: Case of 15 sites and 5 allowed queries. The table shows robot and human performance for each site, as well as the prediction of human performance.

3.5.2 Further Evaluation over 15 Sites

In order to demonstrate the performance at a larger scale, we next show simulation results where a robot is tasked with finding humans in 15 sites with limited help from an operator. More specifically, it is given a maximum of 5 queries to operators and can also choose to visit a site after its initial sensing. All the sites have the same cost to visit. The images taken by the robot after its initial sensing are real data taken around our campus and have different levels of difficulty. The final evaluation of the performance is done by passing the chosen images to MTurk users. Table 3.6 summarizes task difficulty of the sites for both robot and human. Without loss of generality, we numerate the sites from hard to easy based on true human performance from 50 MTurk users (p_h). It can be seen that sites 1-5 are hard for both the robot and the human, sites 6-10 are hard for the robot but easy for human, and sites 11-15 are easy for both. The table further shows that our predictor correctly identifies the sites that are hard/easy for humans. Table ?? then compares the decisions of our approach with the benchmark. As can be seen, the two methods result in very different decisions in terms of which sites to visit or query the human operator. The last column shows the average (averaged over queried sites) probability that human operators accomplish the task when asked, based on passing each image to 50 MTurk users. This probability is 0.98 for our approach and 0.58 for the benchmark.

It can be seen that by using our approach, the images selected to query the operator are those that the human operator can indeed be of help with. Without a proper predic-

tion, however, the robot can send several hard images to the operator, instead of further sensing, thus wasting questions and incurring large performance loss.

In this chapter, we developed a DCNN-based approach to predict human visual performance and demonstrated its efficacy in several real-world experiments involving human-robot visual collaboration. In the next chapter, we consider a more general optimization of human-robot collaborations which involve various realistic factors commonly-seen in robotic operations, such as imperfect sensing quality, realistic wireless communication model, realistic robot motion power model, and resource constraints. We develop a framework to co-optimize the robot's field decisions under these realistic factors, while taking into account the predicted human performance.

Chapter 4

Optimizing Human-Robot Collaboration with Human Performance Prediction

In this chapter, we take a more comprehensive look into the optimization of the robot's field decisions, given the predicted human performance. More specifically, we consider a realistic and general collaborative site inspection scenario where there is a pre-defined path, near which there is a number of sites containing targets to be classified, as shown in Fig. 4.1. As the robot traverses this path and reaches a point close to a site, it has to decide whether (and to what extent) it should incur motion energy to deviate from the main path towards the site to sense it better, and whether it should ask for human help with this site. The robot then returns to the pre-defined path after a possible deviation. The robot's objective is then to maximize its target classification performance at all the sites, under limited onboard energy constraints (including both communication and motion), with a limited access to a human operator to ask for help, and while considering the fact that human visual performance is not perfect and varies depending on the sensory input. Furthermore, the robot has constraints in communication with

Parts of this chapter have appeared in our paper [106], ©[2019] IEEE.

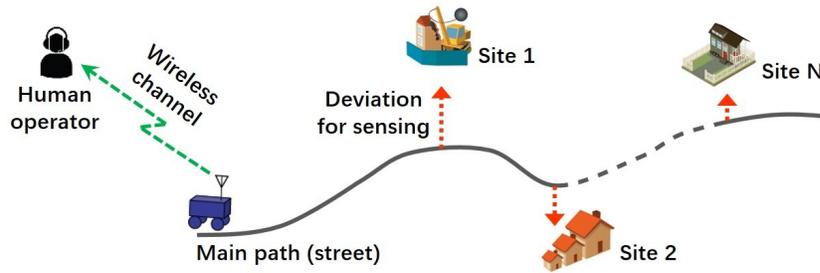


Figure 4.1: The robot travels along a pre-defined path to inspect sites nearby. For each site, the robot decides how much it should deviate from the main path to move closer to the site for better sensing. It also needs to decide whether to ask the remote operator for help with target classification for this site. Human visual performance, however, is not perfect, and can vary depending on the sensing quality.

the human operator. We consider two realistic communication constraints. In the first case, the robot is given a limited number of chances to query the human operator. This case realistically considers the impact of possible human work overload, which needs to be prevented by limiting the number of queries. In the second case, the quality of the robot’s wireless communication to the remote human varies spatially, due to path loss, shadowing, and multipath fading in realistic channel environments. The robot then needs to first predict the channel quality at unvisited locations in the environment and then decide where to communicate with the human operator, given a limited energy budget that can be used towards communication and motion.

The rest of the chapter is organized as follows. In Sec. 4.1 and 4.2, we show how to co-optimize the robot’s motion, sensing, and queries to the operator, considering an imperfect human visual performance, under communication constraints to the operator, and given a limited energy budget. More specifically, in Sec. 4.1, the robot is given a limited number of queries to the operator, while in Sec. 4.2, the quality of communication link to the remote operator may not be perfect everywhere in the workspace. The robot then has a limited energy budget that can be used towards communication and motion.

4.1 Optimizing Human-Robot Collaboration under Total Query Constraints

In this section, we consider the case that the robot has been given a limited number of queries for human help during its operation. As discussed earlier, this represents many realistic situations where the operator has to work with a number of robots and can thus not be overloaded with too many queries. The robot then has to optimally choose the sites for which humans have a high chance of accomplishing the visual task to query the operator. Furthermore, it has to co-plan its trajectory for site inspection, by considering its motion energy budget, its query budget, and human visual performance. In this section, we show how the robot can optimally achieve this.

4.1.1 Problem Setup

Consider the case where a robot travels along a pre-defined main path and has a total number of N sites to inspect, as shown in Fig. 4.1. In order to achieve a good target classification performance, the robot has to decide how far it should deviate from the pre-defined path to sense each site (it returns to the main path after each deviation), and whether it should ask for human help for each site. The robot is allowed to query the human M times during this operation and has a motion energy budget \mathcal{E}_{\max} for the deviations, which is in addition to the motion energy required to travel the main path. We assume that the distance of each site to the pre-defined path is known such that motion energy costs can be estimated.

Let $d_i \in [0, d_{\max,i}]$ denote the deviation distance from the path to site i , $\forall i \in \{1, \dots, N\}$, where $d_{\max,i}$ is the distance between the path and site i . $\mathcal{E}_{m,i}(d_i)$ denotes the motion energy cost of a deviation of d_i , which includes both moving to the site and

returning to the main path. Based on the sensing quality after traveling d_i , the correct classification probabilities of site i are $p_{r,i}(d_i)$ and $p_{h,i}(d_i)$ for the robot and the human, respectively.¹

In order to maximize the average correct classification probability of the sites, we have the following optimization problem:

$$\begin{aligned}
& \max_{\tilde{\gamma}, d} && \frac{1}{N} \sum_{i=1}^N \gamma_i p_{h,i}(d_i) + (1 - \gamma_i) p_{r,i}(d_i) \\
& \text{s.t.} && \sum_{i=1}^N \mathcal{E}_{m,i}(d_i) \leq \mathcal{E}_{\max}, \quad \sum_{i=1}^N \gamma_i \leq M, \\
& && \tilde{\gamma} \in \{0, 1\}^N, \quad x \succeq \mathbf{0},
\end{aligned} \tag{4.1}$$

where $\tilde{\gamma} = [\gamma_1, \dots, \gamma_N]^T$, $d = [d_1, \dots, d_N]^T$. $\gamma_i = 1$ indicates that the robot should query the human for site i and $\gamma_i = 0$ denotes otherwise, and $\mathbf{0}$ is a vector of all 0s.

This optimization problem, however, is in general a non-convex Mixed Integer Non-linear Program, solving which is computationally expensive [115]. We then show that by discretizing the motion decision space, this problem can be formulated as a Multiple-Choice Multidimensional Knapsack Problem (MMKP), which can be further closely approximated by solving a Linear Program (LP).

4.1.2 MMKP Formulation via Motion Space Discretization

In this section, we restrict the robot's deviation to quantized steps, which limits the robot's motion decision to a finite set. Let $\Omega_{d,i} = \{d_{i,j} : j \in \{1, \dots, D_i\}\}$ denote the set of possible motion steps for site i , where $d_{i,j}$ represents a deviation distance corresponding to the j^{th} step and D_i is the number of possible steps for site i . The overall decision set

¹We discuss how the robot can estimate the human's and its own visual classification performance as a function of the sensing distance during operation in Sec. 4.1.7.

for site i then becomes $\Omega_{d,i} \times \{0, 1\}$, where $\{0, 1\}$ is the decision set for whether the robot should ask for human help for site i . This discretization then results in the following Integer Linear Program (ILP):

$$\begin{aligned}
\max_{\gamma} \quad & \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^{D_i} \gamma_{h,i,j} p_{h,i,j} + \gamma_{r,i,j} p_{r,i,j} \\
\text{s.t.} \quad & (1) \sum_{j=1}^{D_i} \gamma_{h,i,j} + \gamma_{r,i,j} = 1, \quad \forall i \in \{1, \dots, N\}, \\
& (2) \sum_{i=1}^N \sum_{j=1}^{D_i} (\gamma_{r,i,j} + \gamma_{h,i,j}) \mathcal{E}_{m,i,j} \leq \mathcal{E}_{\max}, \\
& (3) \sum_{i=1}^N \sum_{j=1}^{D_i} \gamma_{h,i,j} \leq M, \quad (4) \gamma \in \{0, 1\}^{2 \times \sum_{i=1}^N D_i},
\end{aligned} \tag{4.2}$$

where γ is the stacked vector containing all the optimization variables as defined by Eq (4.3), with N denoting the number of sites and D_i representing the number of motion decisions of site i . More specifically, for site i , the set of decision variables is $\{\gamma_{r,i,1}, \gamma_{h,i,1}, \dots, \gamma_{r,i,D_i}, \gamma_{h,i,D_i}\}$, where $\gamma_{r,i,j} = 1$ if the robot deviates $d_{i,j}$ from the path and relies on itself for classification and $\gamma_{r,i,j} = 0$ denotes otherwise. Similarly, $\gamma_{h,i,j} = 1$ if the robot deviates $d_{i,j}$ and queries the human *after* the further sensing and $\gamma_{h,i,j} = 0$ denotes otherwise. γ is then the stacked vector of all the decision variables for all the sites. Note that due to constraint (3), for each site, only one variable will be equal to 1 and all the others will be 0 in a feasible solution. After the discretization, we represent $p_{h,i}(d_{i,j})$, $p_{r,i}(d_{i,j})$, and $\mathcal{E}_{m,i}(d_{i,j})$ by $p_{h,i,j}$, $p_{r,i,j}$, and $\mathcal{E}_{m,i,j}$ for conciseness, which are non-negative

constants.

$$\begin{aligned} \gamma = & \underbrace{[\gamma_{r,1,1}, \gamma_{h,1,1}, \dots, \gamma_{r,1,D_1}, \gamma_{h,1,D_1}, \dots,}_{\text{variables for site 1}} \\ & \underbrace{\gamma_{r,i,1}, \gamma_{h,i,1}, \dots, \gamma_{r,i,D_i}, \gamma_{h,i,D_i}, \dots,}_{\text{variables for site } i} \\ & \underbrace{\gamma_{r,N,1}, \gamma_{h,N,1}, \dots, \gamma_{r,N,D_N}, \gamma_{h,N,D_N}]^T}_{\text{variables for site } N} \end{aligned} \quad (4.3)$$

The next proposition shows that problem (4.2) is an MMKP by comparing its structure and parameters to those of a standard MMKP formulation.²

Proposition 4.1 *Problem (4.2) is an MMKP, where there are N classes (sites) and for each class, there is a total number of $2 \times D_i$ items (decisions). Furthermore, there are two resource constraints, which correspond to the motion energy budget and the query budget.*

Proof: This can be established by comparing problem (4.2) with problem (A.1). ■

In the following proposition, we characterize the approximation error due to solving the discretized problem (4.2) instead of the original continuous problem (4.1).

Proposition 4.2 *Denote the optimums of problems (4.1) and (4.2) as f_{Cont}^* and f_{MMKP}^* , respectively. Then, we have $f_{Cont}^* - f_{MMKP}^* \in [0, \tilde{g}]$, where $\tilde{g} = \max\{|p_{h,i,j+1} - p_{h,i,j}|, |p_{h,i,j+1} - p_{r,i,j}|, |p_{r,i,j+1} - p_{h,i,j}|\}, \forall i \in \{1, \dots, N\}$ and $j \in \{1, \dots, D_i\}$.*

Proof: Since an optimal solution to problem (4.2) is also feasible to problem (4.1), we have $f_{Cont}^* \geq f_{MMKP}^*$. Suppose that for each site i , the motion cost $\mathcal{E}_{m,i}(d_i)$ is non-decreasing in the deviation distance d_i . Denote an optimal solution to problem (4.1) by $\tilde{\gamma}^*$ and d^* . For each site i , to obtain a feasible solution to problem (4.2), we replace

²See Appendix A for an overview of MMKP.

d_i^* with $d_{i,j}^* = \operatorname{argmin}|d_{i,j} - d_i^*|$, where $d_{i,j} \leq d_i^*$, $d_{i,j} \in \Omega_{d,i}$, and $\Omega_{d,i}$ is the discretized motion decision set for site i . Then, in the feasible solution to problem (4.2), for site i , the decision is represented by a deviation distance of $d_{i,j}^*$ and a human query indicator γ_i^* . It can then be easily confirmed that the performance degradation is upper bounded by g for each site and thus, we have $f_{\text{Cont}}^* - f_{\text{MMKP}}^* \leq \tilde{g}$. ■

Based on Prop. 4.2, it can be seen that if a fine-grained discretization is used, the difference between the optimums of problem (4.1) and problem (4.2) is negligible. However, a fine-grained discretization will introduce a large number of variables in the MMKP. Since an MMKP is an NP-hard combinatorial problem, it is not computationally efficient to solve a large-scale MMKP. Therefore, in the next section, we show how to efficiently obtain a solution based on LP relaxation and further prove that the LP-based solution is very close to the optimums of problems (4.1) and (4.2).

4.1.3 Near-Optimal LP-based Solution

In this section, we propose an approach to efficiently obtain a near-optimal solution to problem (4.2) based on its LP relaxation.³

The LP relaxation is obtained by replacing the binary constraint $\gamma \in \{0, 1\}^{2 \times \sum_{i=1}^N D_i}$ with a linear constraint $\mathbf{0} \preceq \gamma \preceq \mathbf{1}$, where “ \preceq ” denotes a component-wise comparison, and $\mathbf{0}$ and $\mathbf{1}$ are vectors of all 0s and all 1s, respectively. We start this section by studying some properties of any feasible solution to this LP relaxation of problem (4.2). Based on these properties, we then characterize the number of fractional (non-binary) variables in an optimal solution to the LP relaxation of problem (4.2). We further prove that by properly rounding the LP optimal solution (to achieve a binary solution), we can achieve a solution that is very close to the optimum of problem (4.2), as well as problem (4.1).

³Note that the LP relaxation of an ILP results in the same optimal solution as its Lagrangian relaxation [116].

To the best of our knowledge, this is the first proof characterizing the number of fractional variables in an optimal solution to the LP relaxation of an MMKP when there is more than one resource constraint.

Consider an initial feasible solution to the LP relaxation of problem (4.2). We first describe a method to modify this solution to obtain another feasible solution. In other words, the variables in the modified solution still satisfy all the constraints of the LP. Suppose that in a feasible solution to the LP relaxation of problem (4.2), there is a number of fractional variables: $\gamma_{h,i,j}$ with $(i,j) \in S_H = \{(i,j) : \gamma_{h,i,j} \notin \{0,1\}\}$ and $\gamma_{r,k,m}$ with $(k,m) \in S_R = \{(k,m) : \gamma_{r,k,m} \notin \{0,1\}\}$. We introduce some small changes to the fractional variables:

$$\widehat{\gamma}_{h,i,j} = \gamma_{h,i,j} + \delta\alpha_{i,j} \quad \text{and} \quad \widehat{\gamma}_{r,k,m} = \gamma_{r,k,m} + \delta\beta_{k,m}, \quad (4.4)$$

where $|\delta| > 0$ is a small number, and $\alpha_{i,j}, \beta_{k,m} \in \mathcal{R}$ are numbers to be determined. The symbol “ $\widehat{}$ ” denotes the modified solution. The non-fractional variables stay the same.

Let α denote a stacked vector of $\alpha_{i,j}$ with $(i,j) \in \Omega_H$ and β denote a stacked vector of $\beta_{k,m}$ with $(k,m) \in \Omega_R$. In addition to requiring the modified solution to satisfy the optimization constraints of the LP, we also enforce that the modified solution uses the same amount of resources as the initial solution. Then, α and β should satisfy the following system of linear equations:

$$\begin{aligned} (1) \quad & \sum_{(i,j) \in \Omega_H} \alpha_{i,j} + \sum_{(i,m) \in \Omega_R} \beta_{i,m} = 0, \quad \forall i \in \Omega_f \\ (2) \quad & \sum_{(i,j) \in \Omega_H} \alpha_{i,j} = 0, \\ (3) \quad & \sum_{(i,j) \in \Omega_H} \mathcal{E}_{i,j} \alpha_{i,j} + \sum_{(k,m) \in \Omega_R} \mathcal{E}_{k,m} \beta_{k,m} = 0, \end{aligned} \quad (4.5)$$

where Ω_f is the set of sites with fractional variables, line 1 corresponds to constraint (1) in problem (4.2) that the decision variables of each site should add up to 1, lines 2 and 3 ensure that the number of queries used and the motion energy usage do not change after modifying the initially-given feasible solution. It can be seen that if α and β satisfy system (4.5), then constraints (1) - (3) of the LP relaxation of problem (4.2) are satisfied. Additionally, $|\delta| > 0$ can be made sufficiently small such that $\mathbf{0} \preceq \hat{\gamma} \preceq \mathbf{1}$.

We can write system (4.5) in matrix form, $A \cdot [\alpha, \beta]^T = \mathbf{0}$, where A has the following form:

$$\left[\begin{array}{cccc|cccc} \mathbf{1}_{\tilde{D}_{H,1}} & \mathbf{0}_{\tilde{D}_{H,2}} & \cdots & \mathbf{0}_{\tilde{D}_{H,N}} & \mathbf{1}_{\tilde{D}_{R,1}} & \mathbf{0}_{\tilde{D}_{R,2}} & \cdots & \mathbf{0}_{\tilde{D}_{R,N}} \\ \mathbf{0}_{\tilde{D}_{H,1}} & \mathbf{1}_{\tilde{D}_{H,2}} & \cdots & \mathbf{0}_{\tilde{D}_{H,N}} & \mathbf{0}_{\tilde{D}_{R,1}} & \mathbf{1}_{\tilde{D}_{R,2}} & \cdots & \mathbf{0}_{\tilde{D}_{R,N}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{\tilde{D}_{H,1}} & \mathbf{0}_{\tilde{D}_{H,2}} & \cdots & \mathbf{1}_{\tilde{D}_{H,N}} & \mathbf{0}_{\tilde{D}_{R,1}} & \mathbf{0}_{\tilde{D}_{R,2}} & \cdots & \mathbf{1}_{\tilde{D}_{R,N}} \\ \hline \mathbf{1}_{\tilde{D}_{H,1}} & \mathbf{1}_{\tilde{D}_{H,2}} & \cdots & \mathbf{1}_{\tilde{D}_{H,N}} & \mathbf{0}_{\tilde{D}_{R,1}} & \mathbf{0}_{\tilde{D}_{R,2}} & \cdots & \mathbf{0}_{\tilde{D}_{R,N}} \\ & \mathcal{E}_{m,H} & & & & \mathcal{E}_{m,R} & & \end{array} \right] \quad (4.6)$$

where $\mathbf{1}_{\tilde{D}_{H,i}}$ and $\mathbf{0}_{\tilde{D}_{H,i}}$ are $1 \times \tilde{D}_{H,i}$ row vectors of all 1s and all 0s, respectively, and $\mathbf{1}_{\tilde{D}_{R,i}}$ and $\mathbf{0}_{\tilde{D}_{R,i}}$ are $1 \times \tilde{D}_{R,i}$ row vectors of all 1s and all 0s, respectively, $\forall i \in \{1, \dots, N\}$. $\tilde{D}_{H,i}$ and $\tilde{D}_{R,i}$ are the numbers of fractional decision variables of site i , associated with the robot's decisions of querying the human and relying on its own classification, respectively.

The entries of A to the left of the vertical dashed line correspond to α and those to the right correspond to β . The entries above the horizontal dashed line correspond to line 1 in system (4.5). The two bottom rows correspond to lines 2 and 3, respectively, where $\mathcal{E}_{m,H} = [\mathcal{E}_{m,i,j}] \in \mathcal{R}^{1 \times |\Omega_H|}$ is a row vector containing the motion energy costs of $\gamma_{h,i,j}$ with $(i, j) \in \Omega_H$ and $\mathcal{E}_{m,R} = [\mathcal{E}_{m,k,m}] \in \mathcal{R}^{1 \times |\Omega_R|}$ is a row vector containing the motion energy costs of $\gamma_{r,k,m}$ with $(k, m) \in \Omega_R$.

If there exists a non-trivial solution to Eq. (4.5) and $|\delta|$ is sufficiently small, then we

can obtain another feasible solution by modifying the initial feasible solution based on Eq. (4.4). In order for Eq. (4.5) to have non-trivial solutions, matrix A needs to have a non-zero null space. In the following lemma, we formally characterize the conditions under which A has a non-zero null space.

Lemma 4.1 *Consider a feasible solution to the LP relaxation of problem (4.2), and a modification of it to achieve another feasible solution through Eq. (4.4) and (4.5). There exist non-trivial solutions to system (4.5) (non-zero null space for matrix A in Eq. (4.6)) if any one of the following conditions hold:*

- 1) *the fractional variables are associated with more than 2 sites;*
- 2) *the fractional variables are associated with 2 sites and there are more than 2 fractional variables associated with at least one of the two sites;*
- 3) *the fractional variables are associated with 1 site and there are more than 3 fractional variables.*

Proof: 1) It can be seen that there are $n + 2$ rows in A , where n is the number of sites with fractional variables. Since the decision variables associated with each site add up to 1, there must be at least two fractional variables associated with each of these n sites. Thus, there are at least $2n$ columns in A . Then, when $n > 2$, system (4.5) is under-determined, and therefore, A has a non-zero null space.

2) If the fractional variables are associated with exactly 2 sites and there are more than 2 fractional variables with at least one of the sites, then there will be 4 rows and more than 4 columns in A , resulting in a non-zero null space for A .

3) If the fractional variables are associated with 1 site and there are more than 3 fractional variables, then there will be 3 rows and more than 3 columns in A , resulting in a non-zero null space for matrix A . ■

It is easy to confirm that there always exists a sufficiently small $|\delta|$ such that $\widehat{\gamma}_{h,i,j}, \widehat{\gamma}_{r,i,j} \in [0, 1]$, $\forall i \in \{1, \dots, N\}$ and $j \in \{1, \dots, D_i\}$. Then, the existence of non-trivial solutions to system (4.5) and a sufficiently small $|\delta|$ ensure that the modified variables satisfy the constraints of the LP. Denote the objective function values of the initial solution and the modified solution (based on Eq. (4.4) and (4.5)) by f_{LP} and \widehat{f}_{LP} , respectively. Then, f_{LP} and \widehat{f}_{LP} are related as follows:

$$\widehat{f}_{LP} = f_{LP} + \delta\Delta, \quad (4.7)$$

where $\Delta = \sum_{(i,j) \in \Omega_H} \alpha_{i,j} p_{h,i,j} + \sum_{(k,m) \in \Omega_R} \beta_{k,m} p_{r,k,m}$. By applying Lemma (4.1) and analyzing $\delta\Delta$, we then characterize the fractional variables in an optimal solution to the LP relaxation of problem (4.2) in the following proposition.

Proposition 4.3 *There exists an optimal solution, γ_{LP}^* , to the LP relaxation of problem (4.2) that satisfies the following:*

- 1) γ_{LP}^* has at most four fractional (non-binary) variables;
- 2) The fractional variables are associated with at most two sites;
- 3) If the fractional variables are associated with two sites, then two fractional variables are associated with each site; if the fractional variables are associated with only one site, then there exists at most three fractional variables.
- 4) If γ_{LP}^* has no fractional variables, then the solution is also an optimal solution to the MMKP problem (4.2).

Proof: 1) Suppose that we have an optimal solution to the LP relaxation of problem (4.2) with fractional variables. If this optimal solution does not satisfy any of the conditions in Lemma 4.1, then there are at most four fractional variables. On the other hand, if the optimal variables satisfy any of the conditions in Lemma 4.1, then the fractional variables can be modified as described in Eq. (4.4), where α and β are obtained

from a non-trivial solution to system (4.5). With a small $|\delta|$, this modified solution satisfies the constraints of the LP. Denote the objective function values of the optimal LP solution and the modified feasible solution as f_{LP}^* and \widehat{f}_{LP} , respectively. We then have $\widehat{f}_{LP} = f_{LP}^* + \delta\Delta$ based on Eq. (4.7). We discuss two cases based on the value of Δ .

Case 1 ($\Delta = 0$): If $\Delta = 0$, then $f_{LP}^* = \widehat{f}_{LP}$, regardless of the value of δ . This indicates that there exist multiple optimal solutions to the LP and that if we modify the initial optimal solution based on Eq. (4.4) and (4.5), we can obtain another solution which is also optimal. We can then increase $|\delta|$ until at least one of the modified fractional variables becomes 0 or 1, resulting in a strictly smaller number of fractional variables in the solution. As long as the (modified) variables satisfy any one of the conditions in Lemma 4.1, this process can be repeated to reduce the number of fractional variables while maintaining the optimality of the solution. When we cannot perform this modification procedure anymore, we arrive at an optimal solution that does not satisfy any of the conditions in Lemma 4.1. Therefore, in this case, there exists an optimal solution γ_{LP}^* with at most four fractional variables.

Case 2 ($\Delta \neq 0$): If $\Delta \neq 0$, we can choose the sign of δ such that $\delta\Delta > 0$, resulting in $\widehat{f}_{LP} > f_{LP}^*$. This contradicts the assumption that f_{LP}^* is the optimum. Thus, in this case, the optimal solution can not satisfy any of the conditions in Lemma 4.1, and γ_{LP}^* have at most four fractional variables.

In summary, based on the analysis of the two cases above, there exists an optimal solution to the LP which does not satisfy any of the conditions in Lemma 4.1, and thus, has at most four fractional variables.

2) & 3) Based on part 1) of the proof, there exists an optimal LP solution that does not satisfy any of the conditions in Lemma 4.1. Such an optimal solution then also satisfies conditions (2) and (3) of this proposition.

4) As the LP relaxation of problem (4.2) has a larger feasible set, its optimum will be

no worse than that of problem (4.2). If the optimal LP solution consists of only binary variables, then it is also an optimal solution to problem (4.2). ■

Based on Prop. 4.3, the following theorem bounds the optimality gap between a feasible solution to problem (4.2) obtained by rounding the fractional variables (to $\{0, 1\}$) in the optimal LP solution and the optimum of problem (4.2). Our derived bound is general and independent of the rounding technique.

Theorem 4.1 *Based on the optimal solution to the LP relaxation of problem (4.2), a feasible solution can be constructed to problem (4.2), which is less than or equal to $2 \times (p_{\max} - p_{\min})/N$ from the optimum of problem (4.2), where $p_{\max} = \max\{p_{h,i}(d_i), p_{r,i}(d_i)\}$, $p_{\min} = \min\{p_{h,i}(d_i), p_{r,i}(d_i)\}$, $\forall i \in \{1, \dots, N\}$ and $d_i \in [0, d_{\max,i}]$.*

Proof: If the optimal LP solution has no fractional variables, then it is also optimal to problem (4.2). On the other hand, if it has fractional variables, Prop. 4.3 states that it has at most four fractional variables associated with at most two sites. Then, for sites with fractional variables, we round the fractional variables such that the overall query and motion energy constraints are still satisfied, which provides a feasible solution to problem (4.2). For the overall rounding of a site, the performance degradation is at most $p_{\max} - p_{\min} = \max\{p_{h,i}(d_i), p_{r,i}(d_i)\} - \min\{p_{h,i}(d_i), p_{r,i}(d_i)\} \leq 1$, where $i \in \{1, \dots, N\}$ and $d_i \in [0, d_{\max,i}]$, independent of the rounding method. Denote the optimums of the LP and the MMKP, and the objective function value of the LP-based solution obtained by rounding as f_{LP}^* , f_{MMKP}^* and $f_{\text{LP-based}}$. We then have $f_{\text{LP}}^* \geq f_{\text{MMKP}}^* \geq f_{\text{LP-based}}$. Since there are at most two sites in the optimal LP solution that require rounding, we have $f_{\text{LP}}^* - f_{\text{LP-based}} \leq 2 \times (p_{\max} - p_{\min})/N$, which further gives $f_{\text{MMKP}}^* - f_{\text{LP-based}} \leq 2 \times (p_{\max} - p_{\min})/N$. ■

Theorem 4.1 provides a general bound that applies to any form of rounding the fractional variables, while satisfying the resource constraints. Next, we present a specific

method of rounding the fractional variables in Alg. 2 that optimizes the performance. For the rest of this section, we refer to this feasible solution to problem (4.2) given by Alg. 2 as the *LP-based solution*, which will be used in our numerical studies.

In Alg. 2, we denote γ_{LP}^* as the optimal solution to the LP relaxation of problem (4.2) and $\gamma_{\text{LP-based}}$ as the LP-based feasible solution to problem (4.2) obtained via the rounding in Alg. 2. Similar to problem (4.2), subscripts “ r, i, j ” (“ h, i, j ”) indicate that the robot will deviate $d_{i,j}$ towards the site, and then rely on its own classification (query the operator). In the optimal LP solution γ_{LP}^* , Ω_f is the set of sites with fractional variables, M_f is the total number of queries allocated to the fractional variables: $\sum_{i \in \Omega_f} \sum_{j=1}^{D_i} \gamma_{\text{LP},h,i,j}^*$, and \mathcal{E}_i is the energy allocated to site i : $\sum_{j=1}^{D_i} (\gamma_{\text{LP},r,i,j}^* + \gamma_{\text{LP},h,i,j}^*) \mathcal{E}_{m,i,j}$.

Given the optimal LP solution, Alg. 2 optimizes the decisions over the sites with fractional variables,⁴ such that the resulting solution contains only binary variables, and the energy consumption per site is kept the same as in the LP solution. For instance, in case 2, there is one site with fractional variables ($|\Omega_f| = 1$) in the LP solution (denoted as site i). We then keep the same energy consumption of this site and solve for the best binary decision for site i , as follows. If the total number of queries allocated to the fractional variables is less than one ($M_f < 1$), then we cannot query site i . In that case, we choose the position in site i with the best robot classification performance, subject to the motion energy consumption of site i (\mathcal{E}_i), and set the corresponding variable to one. In this manner, the robot obtains an optimized binary decision for site i , using the same or less resource as allocated by the LP solution to site i . On the other hand, if $M_f \geq 1$, we first look at the two positions with the highest probabilities of human performance (p_{h,i,j_h^*}) and robot performance (p_{r,i,j_r^*}), respectively, given the energy consumption of this site. If $p_{h,i,j_h^*} \geq p_{r,i,j_r^*}$, then the robot should move to position j_h^* and ask about the

⁴If a site has fractional variables, then all the variables of this site must be strictly less than 1 due to constraint (1) of Eq. (4.2).

Algorithm 2: LP-based Solution to the MMKP Problem (4.2)

CASE 1 ($|\Omega_f| = 0$): Set $\gamma_{\text{LP-based}} = \gamma_{\text{LP}}^*$.

CASE 2 ($|\Omega_f| = 1$): Suppose that site i has fractional variables.

Set $j_r^* = \operatorname{argmax}\{p_{r,i,j} : \mathcal{E}_{m,i,j} \leq \mathcal{E}_i\}$.

Set $j_h^* = \operatorname{argmax}\{p_{h,i,j} : \mathcal{E}_{m,i,j} \leq \mathcal{E}_i\}$.

if $M_f < 1$ **then**

 | Set $\gamma_{\text{LP-based},r,i,j_r^*} = 1$.

else

 | If $p_{r,i,j_r^*} \geq p_{h,i,j_h^*}$, set $\gamma_{\text{LP-based},r,i,j_r^*} = 1$. Otherwise, set $\gamma_{\text{LP-based},h,i,j_h^*} = 1$.

end

Set the other variables associated with site i to 0. For the remaining sites, set the variables in $\gamma_{\text{LP-based}}$ equal to their corresponding ones in γ_{LP}^* .

CASE 3 ($|\Omega_f| = 2$): Suppose that sites i_1 and i_2 have fractional variables.

Set $j_r^* = \operatorname{argmax}\{p_{r,i_1,j} : \mathcal{E}_{m,i_1,j} \leq \mathcal{E}_{i_1}\}$.

Set $j_h^* = \operatorname{argmax}\{p_{h,i_1,j} : \mathcal{E}_{m,i_1,j} \leq \mathcal{E}_{i_1}\}$.

Set $k_r^* = \operatorname{argmax}\{p_{r,i_2,k} : \mathcal{E}_{m,i_2,k} \leq \mathcal{E}_{i_2}\}$.

Set $k_h^* = \operatorname{argmax}\{p_{h,i_2,k} : \mathcal{E}_{m,i_2,k} \leq \mathcal{E}_{i_2}\}$.

if $M_f < 1$ **then**

 | Set $\gamma_{\text{LP-based},r,i_1,j_r^*} = 1$ and $\gamma_{\text{LP-based},r,i_2,k_r^*} = 1$.

else if $M_f \geq 2$ **then**

 | For site i_1 , set the variable corresponding to the larger element of $\{p_{r,i_1,j_r^*}, p_{h,i_1,j_h^*}\}$ to 1. For site i_2 , set the variable corresponding to the larger element of $\{p_{r,i_2,k_r^*}, p_{h,i_2,k_h^*}\}$ to 1.

else

if $p_{h,i_1,j_h^*} - p_{r,i_1,j_r^*} \geq p_{h,i_2,k_h^*} - p_{r,i_2,k_r^*} \geq 0$ **then**

 | Set $\gamma_{\text{LP-based},h,i_1,j_h^*} = 1$ and $\gamma_{\text{LP-based},r,i_2,k_r^*} = 1$.

else if $p_{h,i_2,k_h^*} - p_{r,i_2,k_r^*} \geq p_{h,i_1,j_h^*} - p_{r,i_1,j_r^*} \geq 0$ **then**

 | Set $\gamma_{\text{LP-based},r,i_1,j_r^*} = 1$ and $\gamma_{\text{LP-based},h,i_2,k_h^*} = 1$.

else

 | For site i_1 , set the variable corresponding to the larger one of $\{p_{r,i_1,j_r^*}, p_{h,i_1,j_h^*}\}$ to 1. For site i_2 , set the variable corresponding to the larger one of $\{p_{r,i_2,k_r^*}, p_{h,i_2,k_h^*}\}$ to 1.

end

end

Set the other variables associated with site i_1 and i_2 to 0. For the remaining sites, set the variables in $\gamma_{\text{LP-based}}$ equal to their corresponding ones in γ_{LP}^* .

site. Otherwise, the robot should move to position j_r^* and rely on itself. Similarly, in case 3, we solve for the best binary decisions for the sites with fractional variables, while ensuring that the motion energy consumption per site does not increase and the query budget is satisfied. Note that one cannot simply round the fractional variables to the closest integers, as this may violate the given resource constraints.

Next, we characterize the optimal gap between the LP-based solution to problem (4.2) and the optimum of the original continuous problem (4.1).

Corollary 4.1 *The LP-based solution to problem (4.2) is less than or equal to $2 \times (p_{\max} - p_{\min})/N + \tilde{g}$ from the optimum of problem (4.1), where \tilde{g} is the approximation error bound introduced in Prop. 4.2.*

Proof: Based on Prop. 4.2 and Theorem 4.1, we have $f_{\text{Cont}}^* - f_{\text{MMKP}}^* \leq \tilde{g}$ and $f_{\text{MMKP}}^* - f_{\text{LP-based}} \leq 2 \times (p_{\max} - p_{\min})/N$. Thus, $f_{\text{Cont}}^* - f_{\text{LP-based}} \leq 2 \times (p_{\max} - p_{\min})/N + \tilde{g}$ ■

Although the LP relaxation-based solution does not always provide the exact optimal solution to problem (4.2), it offers a very close approximation. For instance, as Theorem 4.1 shows, the gap between the MMKP optimum and the LP-based feasible solution is very small, especially when N is large. Furthermore, unlike the NP-hard MMKP problem (4.2), solving the LP is computationally efficient. By choosing a fine-grained motion discretization, we can then get a near-optimal efficient solution to the original difficult continuous problem (4.1).

Remark 4.1 *(Optimality gap for general MMKPs) Our analysis can be extended to a general MMKP. More specifically, based on a similar analysis to that of Lemma 4.1 and Prop. 4.3, it can be confirmed that in an optimal solution to the LP relaxation of a general MMKP with N_R resource constraints, there exist at most $2 \times N_R$ fractional*

variables and the fractional variables are associated with at most N_R classes. Based on similar arguments to those in Theorem 4.1, the optimality gap between the N_R -resource MMKP, whose objective function is the average reward from each class, and its LP-based solution is upper bounded by $N_R \times (p_{max} - p_{min})/N$, where p_{max} and p_{min} are the maximum and minimum rewards associated with a variable, respectively. We skip the proof due to the space limitation.

4.1.4 Properties of the Optimal Solution

Next, we study some properties of the optimal decisions.

Proposition 4.4 *In an optimal solution to problem (4.2), if $\gamma_{h,i_1,j_1}^* = 1$ and $\gamma_{r,i_2,j_2}^* = 1$, where $i_1, i_2 \in \{1, \dots, N\}$, $j_1 \in \{1, \dots, D_{i_1}\}$, $j_2 \in \{1, \dots, D_{i_2}\}$, and $\max\{d_{i_1,j_1}, d_{i_2,j_2}\} < \min\{d_{max,i_1}, d_{max,i_2}\}$, then $p_{h,i_1,j_1} - p_{r,i_1,j_1} \geq p_{h,i_2,j_2} - p_{r,i_2,j_2}$.*

Proof: Suppose that in an optimal solution, $\gamma_{h,i_1,j_1}^* = 1$, $\gamma_{r,i_2,j_2}^* = 1$, and $p_{h,i_1,j_1} - p_{r,i_1,j_1} < p_{h,i_2,j_2} - p_{r,i_2,j_2}$. By letting $\gamma_{h,i_1,j_1} = 0$, $\gamma_{r,i_1,j_1} = 1$, $\gamma_{h,i_2,j_2} = 1$, and $\gamma_{r,i_2,j_2} = 0$, we obtain a strictly better solution using the same amount of motion energy and the same number of queries, which is a contradiction. ■

This proposition says that if we have two sites i_1 and i_2 , for which the robot will ask the human and rely on its own classification, respectively, then there should be a greater benefit from asking for human help for the first site.

Proposition 4.5 *In an optimal solution to problem (4.2), if $\gamma_{r,i_1,j_1}^* = 1$ and $\gamma_{r,i_2,j_2}^* = 1$, where $i_1, i_2 \in \{1, \dots, N\}$, $j_1 \in \{1, \dots, D_{i_1}\}$, $j_2 \in \{1, \dots, D_{i_2}\}$, and $\max\{d_{i_1,j_1}, d_{i_2,j_2}\} < \min\{d_{max,i_1}, d_{max,i_2}\}$, then $p_{r,i_1,j_1} - p_{r,i_1,j_2} \geq p_{r,i_2,j_1} - p_{r,i_2,j_2}$.*

Proof: Suppose that in an optimal solution, $\gamma_{r,i_1,j_1}^* = 1$, $\gamma_{r,i_2,j_2}^* = 1$, and $p_{r,i_1,j_1} - p_{r,i_1,j_2} < p_{r,i_2,j_1} - p_{r,i_2,j_2}$. Then by letting $\gamma_{r,i_1,j_1} = 0$, $\gamma_{r,i_1,j_2} = 1$, $\gamma_{r,i_2,j_1} = 1$, and $\gamma_{r,i_2,j_2} = 0$,

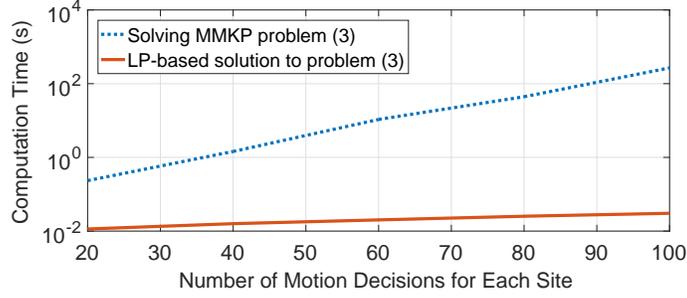


Figure 4.2: Computation time of solving problem (4.2) and obtaining its LP-based solution for 10 sites w.r.t. the number of motion decisions for each site. The computation time is plotted in log scale.

we obtain a strictly better solution using the same amount of motion energy and the same number of queries, which is a contradiction. ■

Proposition 4.6 *In an optimal solution to problem (4.2), if $\gamma_{h,i_1,j_1}^* = 1$ and $\gamma_{h,i_2,j_2}^* = 1$, where $i_1, i_2 \in \{1, \dots, N\}$, $j_1 \in \{1, \dots, D_{i_1}\}$, $j_2 \in \{1, \dots, D_{i_2}\}$, and $\max\{d_{i_1,j_1}, d_{i_2,j_2}\} < \min\{d_{\max,i_1}, d_{\max,i_2}\}$, then $p_{h,i_1,j_1} - p_{h,i_1,j_2} \geq p_{h,i_2,j_1} - p_{h,i_2,j_2}$.*

Proof: The proof is similar to the proof of Prop. 4.5. ■

Props. 4.5 and 4.6 state that if we have two sites i_1 and i_2 , at which the robot deviates two different distances, d_1 and d_2 , respectively, then the performance gain by moving from d_2 to d_1 at site i_1 should be greater than that at site i_2 .

4.1.5 Computation Time

In this section, we numerically compare the computational efficiency of solving the MMKP of problem (4.2) and obtaining its LP-based solution. In this comparison, there are 10 sites, 5 given queries, and the motion energy budget is 50% of what is needed to reach all the sites. The computations are performed on Matlab with an i7 CPU running at 3.4 GHz and the reported computation time is averaged over 500 problem instances.

Fig. 4.2 shows the computations of solving problem (4.2) and obtaining its LP-based solution w.r.t. the number of discretized motion steps per site. It can be seen that the LP-based solution is considerably more efficient. For instance, when a fine-grained discretization of 100 motion decisions per site is used, it takes an average of 268.23 s (or 4.47 min) to compute the MMKP solution. This makes it impractical to directly solve problem (4.2) in a robotics field operation. On the other hand, it takes an average of 0.0304 s to obtain the LP-based solution, making it a more favorable approach to efficiently compute a near-optimal solution in real-life applications where there can be many decision variables.

4.1.6 Online Replanning to Cope with Operation Changes

As can be seen from Theorem 4.1 and Sec. 4.1.5, our proposed approach provides a computationally fast near-optimal solution to problem (4.2). Such computational efficiency enables the robot to quickly replan its actions to cope with possible changes in the operation. For instance, depending on the cognitive load of the human operator, the number of allowed queries may increase or decrease. During the operation, the set of sites which the robot is responsible for inspection may change as well, which also requires an efficient online replanning scheme. Given the updated command, the robot can then quickly recompute its future actions accordingly.

4.1.7 Simulation Results with Real Human Data

We numerically demonstrate the performance of our proposed collaboration framework for target classification under motion energy and query constraints, using real human responses. In order to plan its motion and queries, the robot needs to predict the human's and its own classification performance for the visual data collected at previously-

unvisited sensing locations (e.g., at a location that is $d_{\max,i} - d_{i,j}$ away from site i). In this part, we assume that image noise is the main source of uncertainty for the target classification tasks and utilize an approach that predicts human and robot performance based on the noise variance.⁵ More specifically, we use the Amazon MTurk studies discussed in Sec. 3.1, where we collected 8,000 human visual responses over noisy images. The robot then uses Fig. 3.3 to assess human (as well as its own) visual performance during the operation, by utilizing a sensing-distance-dependent model for the image noise variance.

In the simulations, there are 10 sites, and the distance between each site and the path is 10 m. The noise variance is modeled as a quadratic function of the sensing distance: $v_n = \tilde{a}_i d^2 + \tilde{b}_i$, where \tilde{a}_i and \tilde{b}_i are parameters of the sensing model for site i [41]. We assume that the sites have three levels of sensing difficulty: easy, medium, and hard, encoded in \tilde{a}_i , which is then randomly assigned to the sites. Thus, given a sensing distance, the robot can easily assess the human’s and its own performance as described in Sec. 3.1. The motion model parameters are: $\kappa_1 = 7.4$, $\kappa_2 = 0.29$, and the robot travels with a constant speed of 1 m/s. The motion parameters are based on real power measurements of a Pioneer 3DX robot [2]. The motion energy budget is taken as a percentage of the total energy required to visit all the site locations.

We consider a benchmark methodology where the collaboration is not fully optimized. More specifically, in the benchmark, the robot assumes a perfect human performance (which is a common assumption), but is aware of its own performance (blue dashed curve in Fig. 3.3). The robot then solves problem (4.2) with $p_{h,i,j} = 1$, $\forall i = \{1, \dots, N\}$, $j \in \{1, \dots, D_i\}$.

⁵In practice, the human performance can be affected by a variety of factors, beyond distance, such as image brightness, contrast, resolution, and occlusion. In Sec. 3.2, we have discussed a more comprehensive prediction approach based on a Deep Convolutional Neural Network (DCNN) which can handle any type of uncertainties. As part of future work, one can also incorporate our DCNN-based prediction framework in this path optimization problem by starting with a distance-based prediction and adapting it as the robot gets closer to the sites, by using the more comprehensive DCNN-based prediction.

Next, we compare our proposed approach with the benchmark, by showing the resource savings from using our proposed approach, as compared to the benchmark. More specifically, given any combination of the motion budget and the query budget, we obtain the average correct classification probabilities of our proposed approach and the benchmark, by solving their respective optimization problems, and averaging over 100 problem instances with uniformly random sensing difficulty assignments over the sites. Based on these results, we can then see that to achieve the same average performance, our proposed approach would require smaller resource budgets, as compared to the benchmark.

More specifically, Table 4.1 shows the motion energy saving, enabled by our proposed approach, when the robot is given 6 allowed queries. It can be seen that our proposed approach requires considerably smaller energy budgets than the benchmark to achieve the same performance. For instance, when achieving an average correct classification probability of 0.75, our proposed approach requires a 39.39% smaller motion energy budget. In the table, “Inf” indicates that the benchmark method simply cannot achieve the performance no matter how much motion energy is given. Thus, our proposed approach is able to more efficiently utilize the given energy budget.

Table 4.2 summarizes the saving of queries by using our proposed approach, when the robot is given a motion energy budget of 50% of what is needed to reach all the site locations. It can be seen that our approach requires considerably less communication to the operator. For instance, to achieve an average classification accuracy of 0.85, our approach needs 55.62% fewer queries. This result shows that our proposed approach can reduce the overall burden on human operators who have to operate under fatigue and work overload.

Ave. Correct Classification Prob.	Percentage Energy Saving
0.70	100%
0.75	39.39%
0.80	34.29%
0.85	35.29%
0.90 or higher	Inf

Table 4.1: Motion energy saving by our proposed approach as compared to the benchmark. In this example, the robot is given 6 allowed queries.

Ave. Correct Classification Prob.	Percentage Query Saving
0.80	46.15%
0.85	55.62%
0.90 or higher	Inf

Table 4.2: Saving of queries by our proposed approach as compared to the benchmark. In this example, the motion energy budget is 50% of what is needed to visit all the site locations.

4.2 Optimizing Human-Robot Collaboration under Total Energy Constraints

In the previous section, we considered the case where the communication bottleneck was human work overload, which limited the number of queries the robot could present to the remote operator. In this section, we consider another source of communication bottleneck, the cost of wireless transmission. More specifically, if the robot is operating in the areas where the communication link quality is not good everywhere, and/or if the robot has energy constraints on the communication side, then that can limit the number of successful queries to the human operator. In this section, we *explicitly* model and analyze this case, by considering the true energy cost of wireless transmissions in realistic communication environments that can experience path loss, shadowing, and multipath fading. We further show how the robot can actively take the communication link quality into account when optimizing its collaboration with the operator. More specifically, we show that the resulting human-robot collaborative site survey problem,

under both motion and communication energy constraints, can be posed as a Multiple Choice Knapsack Problem (MCKP) problem. We then derive an efficient near-optimal solution and mathematically characterize several properties of the optimal collaborative solution.

4.2.1 Problem Setup

Consider the setup of Fig. 4.1, where we have a total of N sites. In this section, we consider the case where, instead of a direct limitation on the total number of queries, the robot has a constraint on its total energy consumption, including both communication and motion energy costs. We further consider realistic communication environments, as modeled in the previous part.

Sending a query to the human operator and moving closer to a site for better sensing will incur communication and motion energy costs, respectively. The robot is then given a total energy budget \mathcal{E}_{\max} , which can be used towards communication or motion for site visiting. For the i^{th} site, $i \in \{1, \dots, N\}$, d_i is the deviation distance from the path. We denote the communication energy cost after deviating d_i towards site i by $\mathcal{E}_{c,i}(d_i)$ (i.e., the cost of communicating to the operator from a location that is d_i off the main path towards site i), while $\mathcal{E}_{m,i}(d_i)$ denotes the motion energy cost of deviating a distance of d_i . The probabilities of correct target classification of the i^{th} site, based on the sensing performed after deviating d_i towards site i , are denoted by $p_{r,i}(d_i)$ and $p_{h,i}(d_i)$ for the robot and the human, respectively. In order to maximize the average correct classification probability, the robot needs to decide how far it should deviate from the main path to sense each site, and whether it should ask for help for each site by communicating to the operator after deviating d_i from the main path, subject to a total energy budget. In addition to predicting the human and robot correct classification probabilities, as we discussed in

Sec. 4.1, the robot further probabilistically predicts its communication energy cost, using the formulation of Sec. 2.1, which is needed for a proper optimization.

Similar to in Sec. 4.1, it is possible to formulate this optimization problem as an ILP by discretizing the continuous motion decision space, as follows:

$$\begin{aligned}
\max_{\gamma} \quad & \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^{D_i} \gamma_{h,i,j} p_{h,i,j} + \gamma_{r,i,j} p_{r,i,j} \\
\text{s.t.} \quad & \sum_{j=1}^{D_i} \gamma_{h,i,j} + \gamma_{r,i,j} = 1, \quad \forall i \in \{1, \dots, N\}, \\
& \sum_{i=1}^N \sum_{j=1}^{D_i} \gamma_{h,i,j} \mathcal{E}_{c,i,j} + (\gamma_{r,i,j} + \gamma_{h,i,j}) \mathcal{E}_{m,i,j} \leq \mathcal{E}_{\max}, \\
& \gamma \in \{0, 1\}^{2 \times \sum_{i=1}^N D_i},
\end{aligned} \tag{4.8}$$

where N is the number of sites and D_i is the number of motion decisions of site i . For site i , the set of decision variables is $\{\gamma_{r,i,1}, \gamma_{h,i,1}, \dots, \gamma_{r,i,D_i}, \gamma_{h,i,D_i}\}$, where $\gamma_{r,i,j} = 1$ if the robot deviates $d_{i,j}$ from the path and relies on itself for classification, while $\gamma_{r,i,j} = 0$ denotes otherwise. Similarly, $\gamma_{h,i,j} = 1$ if the robot deviates $d_{i,j}$ from the path and queries the human after further sensing, and $\gamma_{h,i,j} = 0$ denotes otherwise. γ is then the stacked vector of all the decision variables for all the sites, as given by Eq. (4.3). $p_{h,i,j}$, $p_{r,i,j}$, $\mathcal{E}_{c,i,j}$, and $\mathcal{E}_{m,i,j}$ are nonnegative constants representing the probability of human correct classification, the probability of robot correct classification, the communication energy cost of querying the operator, and the motion energy cost, respectively, after deviating $d_{i,j}$ for site i .

As can be seen, the energy-constrained collaborative human-robot site surveillance problem (4.8) is an MCKP, a special form of MMKP, when there is only one type of resource constraint. Due to the prevalence of MCKP problems in computer science, there exists a rich body of literature for mathematically characterizing MCKPs, which we tap

into next for deriving an efficient near-optimal solution to our collaborative problem.

4.2.2 Near-Optimal LP-based Solution

As MCKP is NP-hard and computationally expensive, we show how to utilize an LP relaxation to obtain an efficient near-optimal solution to the MCKP problem (4.8).

Similar to Sec. 4.1.3, the LP relaxation is obtained by replacing the binary constraint $\gamma \in \{0, 1\}^{2 \times \sum_{i=1}^N D_i}$ in problem (4.8) with a linear constraint $\mathbf{0} \preceq \gamma \preceq \mathbf{1}$. The following proposition describes the number of fractional variables in an optimal solution to the LP, which then allows us to relate the LP optimal solution to problem (4.8).

Proposition 4.7 *An optimal solution γ^* to the LP relaxation of the MCKP problem (4.8) satisfies the following:*

- 1) γ^* has at most two fractional variables;
- 2) If γ^* has two fractional variables, then they must be associated with the same site;
- 3) If γ^* has no fractional variables, then the solution is also an optimal solution to the MCKP problem (4.8).

Proof: The proof is similar to that of Prop. 4.3. Alternatively, see the proof of Prop. 2 in [117], which is a different approach based on explicitly constructing the optimal LP solution.⁶ ■

As a consequence of Prop. 4.7, we have the following theorem regarding the optimality gap between the optimum of the MCKP and an LP-based feasible solution.

Theorem 4.2 *Based on the optimal solution to the LP relaxation, a feasible solution can be constructed to problem (4.8), which is less than or equal to $(p_{max} - p_{min})/N$ from the op-*

⁶Note that we can not use the proof of [117] for the MMKP case of Sec. 4.1 as it was developed specifically for the case of one resource constraint (MCKP).

Algorithm 3: LP-based Solution to the MCKP Problem (4.8)

CASE 1 ($|\Omega_f| = 0$): Set $\gamma_{\text{LP-based}} = \gamma_{\text{LP}}^*$.

CASE 2 ($|\Omega_f| = 1$): Suppose that site i has fractional variables.

Set $j_r^* = \operatorname{argmax}\{p_{r,i,j} : \mathcal{E}_{m,i,j} \leq \mathcal{E}_i\}$.

Set $j_h^* = \operatorname{argmax}\{p_{h,i,j} : \mathcal{E}_{m,i,j} + \mathcal{E}_{c,i,j} \leq \mathcal{E}_i\}$.

if $p_{r,i,j_r^*} \geq p_{h,i,j_h^*}$ **then**

 | Set $\gamma_{\text{LP-based},r,i,j_r^*} = 1$.

else

 | Set $\gamma_{\text{LP-based},h,i,j_h^*} = 1$.

end

Set the other variables associated with site i to 0. For the remaining sites, set the variables in $\gamma_{\text{LP-based}}$ equal to the corresponding ones in γ_{LP}^* .

imum of problem (4.8), where $p_{\max} = \max\{p_{h,i}(d_i), p_{r,i}(d_i)\}$, $p_{\min} = \min\{p_{h,i}(d_i), p_{r,i}(d_i)\}$, $\forall i \in \{1, \dots, N\}$ and $d_i \in [0, d_{\max,i}]$.

Proof: The proof is similar to the proof of Theorem 4.1. ■

Alg. 3 next describes our method of rounding the fractional variables in the optimal LP solution, which is similar to Alg. 2 in that while solving for the best binary decisions, Alg. 3 keeps the energy consumption per site the same as in the LP solution. We refer to the feasible solution to problem (4.8), provided by Alg. 3, as the *LP-based solution* for the rest of this section. Similar to Alg. 2, we denote γ_{LP}^* as the optimal solution to the LP relaxation of problem (4.8) and $\gamma_{\text{LP-based}}$ as the LP-based solution to problem (4.8) given by Alg. 3. In the LP optimal solution γ_{LP}^* , Ω_f is the set of sites with fractional variables and \mathcal{E}_i is the energy allocated to site i in the LP optimal solution: $\sum_{j=1}^{D_i} \gamma_{\text{LP},h,i,j}^* \mathcal{E}_{c,i,j} + (\gamma_{\text{LP},r,i,j}^* + \gamma_{\text{LP},h,i,j}^*) \mathcal{E}_{m,i,j}$.

4.2.3 Properties of the Optimal Solution

Next, we study some properties of the optimal decisions.

Proposition 4.8 *In an optimal solution to problem (4.8), if $\gamma_{r,i_1,j_1}^* = 1$ and $\gamma_{r,i_2,j_2}^* =$*

1, where $i_1, i_2 = 1, \dots, N$, $j_1 \in \{1, \dots, D_{i_1}\}$, $j_2 \in \{1, \dots, D_{i_2}\}$, and $\max\{d_{i_1, j_1}, d_{i_2, j_2}\} < \min\{d_{\max, i_1}, d_{\max, i_2}\}$, then $p_{r, i_1, j_1} - p_{r, i_1, j_2} \geq p_{r, i_2, j_1} - p_{r, i_2, j_2}$.

Proof: The proof is through contradiction, similar to the proof of Prop. 4.5. ■

This proposition says that if we have two sites i_1 and i_2 for which the robot relies on its own decision after deviating two different distances towards the sites, d_1 and d_2 respectively, then the performance gain by moving from d_2 to d_1 at site i_1 should be greater than that at site i_2 .

Proposition 4.9 *Consider a simplified scenario where*

1) *The distance between the pre-defined path and each site is the same: $d_{\max, i} = d_{\max}$, $\forall i \in \{1, \dots, N\}$, where $d_m > 0$ is a constant;*

2) *$p_{r, i}(\cdot)$ is the same for all the sites. Similarly, $p_{h, i}(\cdot)$ is the same for all the sites;*

3) *The required communication energy is constant in the vicinity of a site (does not change along the route to a site): $\mathcal{E}_{c, i} = \mathcal{E}_{c, i, 1} = \dots = \mathcal{E}_{c, i, D_i}$, $\forall i = 1, \dots, N$;*

4) *In the route to a site, human and/or robot performance strictly increases as the sensing distance decreases.*

Then, in an optimal solution to problem (4.8), if $\gamma_{h, i_1, j_1}^ = 1$ and $\gamma_{r, i_2, j_2}^* = 1$, where $i_1, i_2 = 1, \dots, N$, $j_1 \in \{1, \dots, D_{i_1}\}$, $j_2 \in \{1, \dots, D_{i_2}\}$, and $\max\{d_{i_1, j_1}, d_{i_2, j_2}\} < \min\{d_{\max, i_1}, d_{\max, i_2}\}$, then $\mathcal{E}_{C, i_1} - \mathcal{E}_{m, \delta} < \mathcal{E}_{c, i_2}$, where $\mathcal{E}_{m, \delta}$ is the motion energy required to move one step closer to a site.*

Proof: Suppose that in an optimal solution, $\gamma_{h, i_1, j_1}^* = 1$, $\gamma_{r, i_2, j_2}^* = 1$, and $\mathcal{E}_{c, i_1} - \mathcal{E}_{m, \delta} \geq \mathcal{E}_{c, i_2}$. Then by letting $\gamma_{h, i_1, j_1} = 0$, $\gamma_{r, i_1, j_2} = 1$, $\gamma_{h, i_2, j_1} = 1$, and $\gamma_{r, i_2, j_2} = 0$, we would obtain the same objective function value with an energy saving of $\mathcal{E}_{c, i_1} - \mathcal{E}_{c, i_2} \geq \mathcal{E}_{m, \delta}$. The robot may then utilize this energy to move one step closer towards any site, resulting in a strictly better solution, which is a contradiction. ■

Consider the case that the discretized motion step is small and thus, the energy cost for moving one step is negligible. Prop. 4.9 then says that, under the conditions in Prop. 4.9, the robot will select the sites with the best communication qualities to query the human operator, as expected.

The following proposition characterizes the optimal solution to the LP relaxation of problem (4.8), where we suppress the subscripts “h” and “r” in the variables as the result does not distinguish between the two cases. More specifically, given a decision variable $\gamma_{i,k}$ for site i , where $k \in \{1, \dots, 2 \times D_i\}$, it can either represent $\gamma_{h,i,j}$ or $\gamma_{r,i,j}$, where $j \in \{1, \dots, D_i\}$. We denote the corresponding human/robot performance and the energy cost of $\gamma_{i,k}$ by $p_{i,k}$ and $\mathcal{E}_{i,k}$, respectively.

Proposition 4.10 *For site i , if $p_{i,j_1} \leq p_{i,j_2} \leq p_{i,j_3}$, $\mathcal{E}_{i,j_1} \leq \mathcal{E}_{i,j_2} \leq \mathcal{E}_{i,j_3}$ and*

$$p_{i,j_2} < \frac{\mathcal{E}_{i,j_2} - \mathcal{E}_{i,j_1}}{\mathcal{E}_{i,j_3} - \mathcal{E}_{i,j_1}} p_{i,j_3} + \frac{\mathcal{E}_{i,j_3} - \mathcal{E}_{i,j_2}}{\mathcal{E}_{i,j_3} - \mathcal{E}_{i,j_1}} p_{i,j_1}, \quad (4.9)$$

then in an optimal solution to the LP relaxation of problem (4.8), $\gamma_{i,j_2}^ = 0$, where $i \in \{1, \dots, N\}$ and $j_1, j_2, j_3 \in \{1, \dots, 2 \times D_i\}$.*

Proof: Suppose that in an optimal solution, $\gamma_{i,j_2}^* = \delta > 0$. It can be confirmed that if Eq. (4.9) holds, then by letting $\gamma_{i,j_2} = 0$, $\gamma_{i,j_1} = \gamma_{i,j_1}^* + \delta(\mathcal{E}_{i,j_3} - \mathcal{E}_{i,j_2})/(\mathcal{E}_{i,j_3} - \mathcal{E}_{i,j_1})$, and $\gamma_{i,j_3} = \gamma_{i,j_3}^* + \delta(\mathcal{E}_{i,j_2} - \mathcal{E}_{i,j_1})/(\mathcal{E}_{i,j_3} - \mathcal{E}_{i,j_1})$, we can further improve the current optimal solution, which is a contradiction. ■

Prop. 4.10 describes the LP-dominance property of the LP relaxation of problem (4.8) [117]. It says that for a site, if a decision’s performance is worse than that of a certain linear combination of two other decisions, then this decision will not be selected in the optimal LP solution.

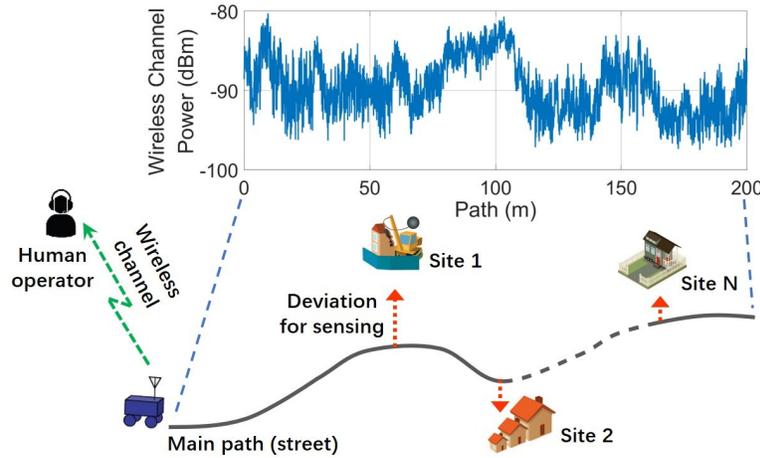


Figure 4.3: The robot travels along a pre-defined main path to inspect sites near the path. Along the path, the robot’s communication quality to the remote operator is subject to a space-varying wireless channel power from downtown San Francisco [3].

4.2.4 Online replanning of Communication and Motion Energy

In our proposed energy-aware collaborative planning, the robot uses its prediction of the communication energy cost for the purpose of planning, which is based on a small number of prior or online channel measurements in the same environment. As the robot spends more time in its workspace during the mission, it can collect more measurements to fine-tune the channel learning (i.e., to better predict communication energy costs). Furthermore, as the robot communicates its sensing data of a site after a site inspection, it will incur the true communication cost at that spot, which may differ from the predicted one. As such, the robot can benefit from replanning its collaborative path planning and decision making, as it obtains more channel samples and updates the cost of communication during the operation. For instance, the robot can re-solve problem (4.8), with the updated available energy resource, after each site visit. While re-solving the MCKP problem (4.8) in an online manner can be computationally expensive, especially when the problem size is large, obtaining the near-optimal LP-based solution to problem (4.8) is fast and efficient, allowing for online replanning, which can be key in

such realistic stochastic settings.

4.2.5 Performance with Real Human Data and Real Wireless Channel Data

In this section, we evaluate our proposed approach using real human data, as described in Sec. 3.1 and also used in the evaluation part of Sec. 4.1. We further use real wireless channel measurements, which were collected along a street in downtown San Francisco [3].⁷ This real data then presents the channel along the main path of the robot, as illustrated in Fig. 4.3. We assume that within the vicinity of a site, the wireless channel is approximately stationary and can be characterized by the same Gaussian random variable. Thus, the expected communication energy required to send a query near a site is constant.⁸ There is a total of 10 sites. We assume that the sites are 10 m off the main street (main path) and 20 m apart. The robot then uses 5% a priori channel measurements to predict the channel quality in this workspace, by using the framework of Sec. 2.1. For the communication system, 64-QAM modulation is used and the transmission time of each query is 2 s, with a target BER of 10^{-6} . The receiver noise power is -100 dBm. For the motion model, we have $\kappa_1 = 7.4$, $\kappa_2 = 0.29$, and the robot travels with a constant speed of 1 m/s. The robot adopts a distance-dependent noise variance model: $v_n = \tilde{a}_i d^2 + \tilde{b}_i$, where \tilde{a}_i and \tilde{b}_i are positive constants [41]. We assume that the sites have three levels of sensing difficulty: easy, medium, and hard, encoded in \tilde{a}_i , which is randomly assigned to the sites. The robot then uses the proposed approach of Sec. 4.2.1 to co-plan its human collaboration and site inspection. Furthermore, after

⁷Channel data is courtesy of W. M. Smith [3].

⁸We make this assumption since this real channel data from downtown San Francisco is only available along a street. As such, we assume the channel remains the same when deviating towards a site. In the next part, we show the performance in a realistic 2D wireless environment where the channel changes along a route to a site as well.

visiting each site, it learns the true channel value and the corresponding true incurred communication energy cost, and can replan its future actions over the remaining sites, using the updated true remaining energy, as discussed in Sec. 4.2.4. Note that after measuring the channel at a site that was planned to be queried, the robot may find that the true required communication energy plus the motion energy to return to the main path will exceed its remaining budget. In this case, it will give up communication for this site and rely on its own classification instead. It then uses the remaining energy to continue replanning for the remaining sites.

Remark 4.2 (*Mission failure without replanning*) *Suppose that the robot does not replan. When the robot communicates to the remote operator at a location with a worse-than-predicted channel quality, it will need to spend more energy than planned to guarantee the required end-to-end communication quality. Since the robot does not replan, it could exhaust the energy budget (\mathcal{E}) and may not be able to arrive at the final destination eventually. Such cases are then considered mission failures. In other words, the no-replanning approach does not always provide a feasible solution, highlighting the importance of replanning.*

Similar to Sec. 4.1.7, we compare our proposed approach with a benchmark methodology where the collaboration is not fully optimized. In this benchmark, the robot assumes that human performance is perfect. The benchmark robot then solves problem (4.8) with $p_{h,i,j} = 1, \forall i = \{1, \dots, N\}, j \in \{1, \dots, D_i\}$.

Fig. 4.4 shows the average correct classification probability of our proposed LP-based solution (Alg. 3) with channel prediction and online replanning (red solid), and compares it to the benchmark with replanning (blue dashed). The performance is averaged over 100 problem instances for each given energy budget. In each instance, the 5% a priori channel measurements are randomly and uniformly chosen over the space, and each site

is randomly and uniformly assigned one of the three sensing difficulty levels. The energy budget is represented as a percentage of what is needed to reach and query all the sites. The figure further shows the performance of solving the MCKP of problem (4.8) with perfect channel knowledge (green dashed), which provides a performance upper bound. As the figure shows, the gap between the upper bound and our approach is considerably small, even though our approach learns the channel with a small number of channel measurements (as opposed to assuming it known), and is based on solving the LP relaxation. It can also be seen that our proposed approach outperforms the benchmark significantly. For instance, given an energy budget of 0.8, the performance upper bound is 0.938, the performance of our proposed LP-based solution with replanning is 0.930, and the performance of the benchmark (with replanning) is 0.785. Note that if the robot does not replan, it may not accomplish the entire mission, as discussed in Remark 4.2. For instance, for the case of our proposed LP-based approach, the robot fails the operation 12.1% of the time if it does not replan, while it will fail 16.0% of the time for the benchmark case without replanning.

Table 4.3 summarizes the energy saving, enabled by our proposed approach, as compared to the benchmark. For instance, our proposed approach can achieve an average correct classification probability of 0.75, but requires 58.46% less energy consumption as compared to the benchmark. “Inf” indicates that the benchmark simply cannot achieve the specified performance no matter how much energy is given.

Fig. 4.5 shows a sample result using our proposed LP-based solution with online replanning. The discretized motion step is 1 m. The base station is located on the far left of the x-axis. The first and second rows show the real and the predicted channel powers along the main path, respectively. The third row then indicates the sensing difficulty associated with each site, with 0 and 2 being the easiest and the hardest, respectively. The fourth row shows the robot’s optimal deviation distance at each site. The fifth row

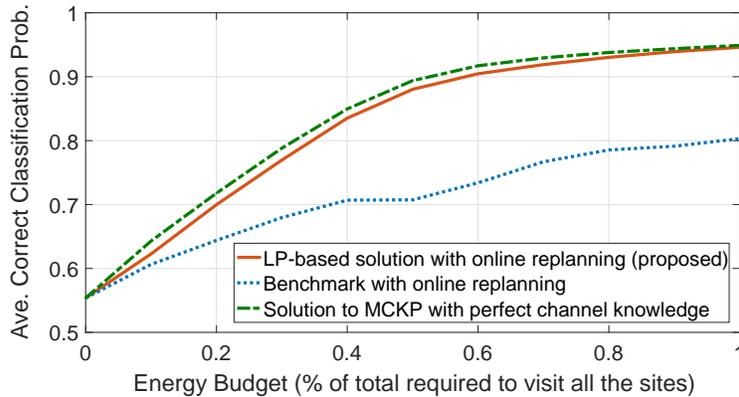


Figure 4.4: The solid red curve shows the performance of our proposed LP-based solution, where the robot has a limited total combined motion and communication energy budget. The dashed blue curve shows the performance of the benchmark. The dashed green curve shows the upper bound performance of solving problem (4.8) with perfect channel knowledge.

Ave. Correct Classification Prob.	Percentage Energy Saving
0.60	27.78%
0.65	37.21%
0.70	46.67%
0.75	58.46%
0.80	64.80%
0.85 or higher	Inf

Table 4.3: Energy saving of our proposed approach as compared to the benchmark, for the case where the robot has a limited total motion and communication energy budget.

indicates whether the robot should ask for human help at each site, with 1 indicating a query.

It can be seen that the robot queries the human operator at sites 5 and 6,⁹ where the channel qualities are the best among all the sites. The channel power at sites 4 and 9 are also good. However, the robot does not query the human operator with these two sites. This is because the sensing difficulty is high for these two sites and the robot needs to move close to the sites for sensing, after which the robot’s own performance is already good and the improvement obtained from asking the human will be small. In

⁹The sites are indexed from 1 to 10 with the leftmost being site 1.

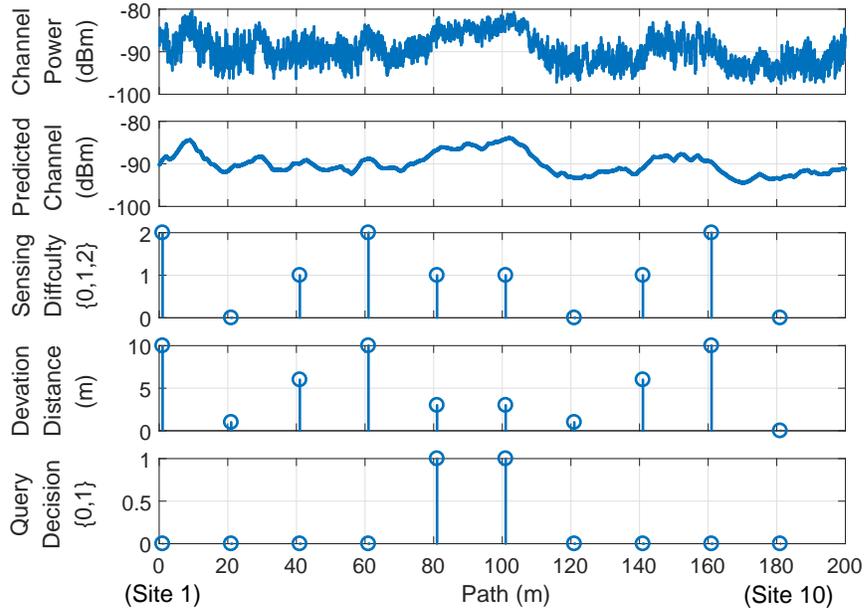


Figure 4.5: The performance of our proposed energy-aware collaborative human-robot site inspection – First row: real channel power along the main path (from downtown San Francisco). Second row: predicted channel power based on 5% prior channel measurements. Third row: the height of the bars indicates the sensing difficulty level, with 0 being the easiest and 2 being the hardest. Fourth row: the height of the bars indicates how much the robot deviates from the path to sense the site. Fifth row: it shows whether the robot queries the human, with 1 indicating a query. The sites are indexed from 1 to 10 with the leftmost being site 1. The energy budget is 30% of what is needed to both reach the site locations and query the human for all the sites.

other words, the robot predicts that the performance gain obtained from querying the human is better for sites 5 and 6 (as compared to sites 4 and 9), and the channel quality is also better at sites 5 and 6. For sites 2, 7, and 10, it can be seen that the robot does not deviate much from the main path for further sensing as it is easy to sense these sites. For sites with a medium sensing difficulty, the robot moves some distance towards the site locations for better sensing (sites 3, 5, 6, and 8). Among these sites, the robot deviates less at sites 5 and 6 where it queries the human. For sites that are hard to sense, the robot needs to move very close to the site locations for a reasonably good sensing quality (site 1, 4, and 9). The average correct classification probability is 0.798 in this case.

4.2.6 Simulation with Realistic 2D Wireless Channel

In this section, we demonstrate our proposed approach in a realistic simulated 2D wireless environment [118], where the channel parameters (obtained from real wireless measurements [1]) are $\hat{\theta}_{ch} = [-41.34, 3.86]$, $\hat{\alpha}_{ch,\text{dB}} = 3.20$, $\hat{\beta}_{ch} = 3.09$ m, and $\hat{\rho}_{ch,\text{dB}} = 1.64$. The sensing model, motion parameters, and communication system parameters are the same as in Sec. 4.2.5. The robot predicts the channel in this workspace based on 0.25% a priori channel samples (randomly located over the workspace), by using the framework of Sec. 2.1. During the operation, the robot utilizes the LP-based approach, with online replanning when optimizing its motion and queries.

For a better illustration, we show an example result with 4 sites in Fig. 4.6. In terms of the sensing difficulty, site 1 is the easiest, site 2 and 4 have a medium sensing difficulty, and site 3 is the hardest to sense. The channel power (in dBm) is plotted on a 2D map with brighter colors (higher values) indicating better channel qualities. The horizontal grey curve represents the pre-defined main path and the orange triangles are the points where the robot deviates from the main path to perform further sensing. The house icons indicate the respective site locations. For each site, the yellow dashed line indicates the extent of the deviation and the presence of a human icon indicates that the robot will query the human operator. For each site, the terminal point of the yellow dashed line shows the location where the robot conducts further sensing and communicates to the remote operator (if the robot decides to query the operator).

It can be seen that for site 1, the robot queries the human operator and stays on the main path. This is because site 1 is easy to sense and the required communication energy is very small for site 1, as the channel quality is very good in this region, which can be seen on the channel power map. In other words, there is no need for further sensing as both the human performance and the channel quality are already good for this site. For

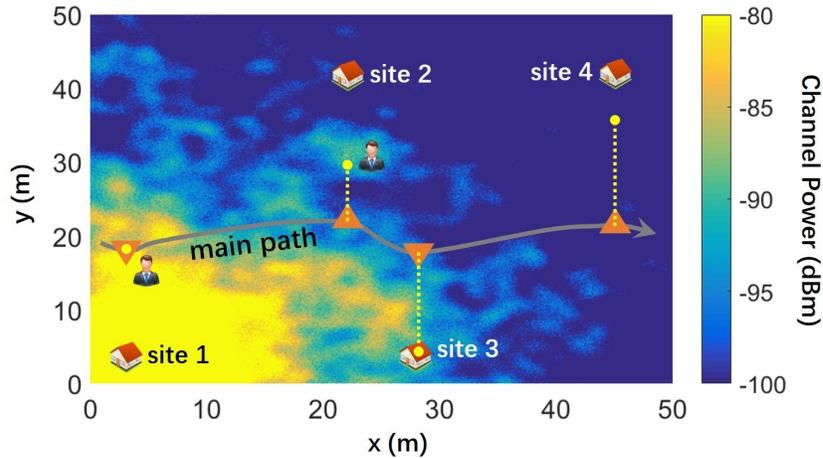


Figure 4.6: Planning result in a 2D wireless environment. Brighter colors (higher values) on the channel map indicate better channel qualities. In this wireless environment, the channel power ranges from -5.40 dBm to -123.06 dBm. For a better visualization, we assign a light yellow color to locations with a channel power greater than -80 dBm and a dark blue color to locations with a channel power less than -100 dBm. The horizontal grey curve represents the pre-defined main path and the orange triangles are the points where the robot deviates from the main path to perform further sensing. The house icons indicate the site locations. For each site, the yellow dashed line indicates the extent of the deviation, and the presence of a human icon indicates that the robot will query the human operator. Readers are referred to the color PDF for optimal viewing.

site 2, the robot deviates some distance and queries the human operator after sensing. In particular, the robot moves to a good channel quality region when communicating to the human, as indicated by the brighter color near the terminal point of the yellow dashed line for site 2. On the other hand, although site 4 also has a medium sensing difficulty, the robot moves much closer to the site location for sensing and does not query the human operator. This is because the channel quality near site 4 is very poor and thus, the robot has to rely on itself for classification by moving close to the site. Site 3 has a high level of sensing difficulty and the robot thus has to reach the site location for sensing. Although the channel quality near site 3 is good, there is no incentive for the robot to query the human operator as its own performance is already good after sensing at the site location. The average correct classification probability is 0.850 in this case.

In this chapter, we considered human-robot collaborative site inspection and target classification, where the human visual performance is not perfect, depending on the sensing quality, and the robot has constraints when communicating with the human operator. In the next chapter, we study another way to improve the robot's visual recognition performance in challenging visual task scenarios, by showing that the output of the onboard DCNN classifier contains robust information on object similarity even if the sensing/classification quality is poor. We then show how to properly exploit such information for path planning and joint labeling.

Chapter 5

Exploiting Object Similarity for Robotic Visual Recognition

In this chapter, we are interested in enhancing the performance of robotic visual classification by exploiting the underlying visual information contained in the output of the robot's onboard Deep Convolutional Neural Network (DCNN) classifier. More specifically, consider the case where robot's initial sensing and scene classification is poor. If the robot can figure out the similarity between the objects in the scene (i.e., which objects belong to the same class), it can then exploit it to improve its classification. But how can the robot infer object similarity, given its poor initial sensing and classification? In this chapter, we show that even when the DCNN classifier cannot properly classify the objects in the field (e.g., due to poor sensing, low illumination, less-seen poses/views), its output can still carry useful information about object similarity. More specifically, we show that the Pearson correlation coefficient of the feature vectors of two object images from the output of a trained DCNN classifier carries key, reliable information on the similarity of the two corresponding objects, in terms of whether they belong to the same class or not. This then allows us to design a Markov Random Field (MRF)-

Parts of this chapter have appeared in our paper [119], ©[2020] IEEE.

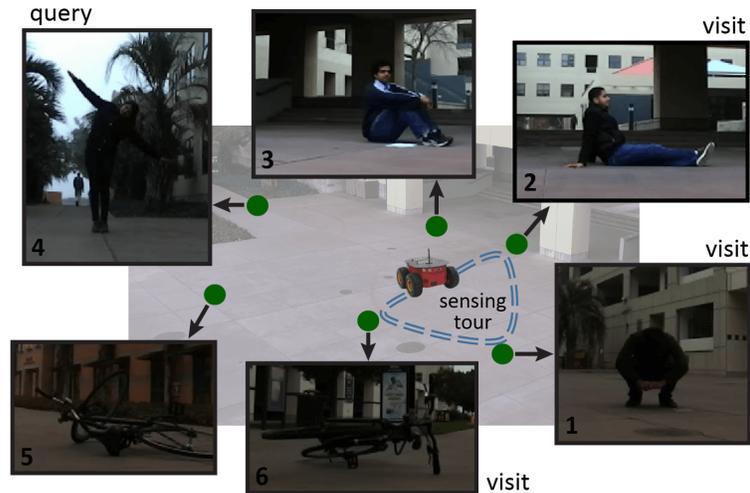


Figure 5.1: A robot is tasked with classifying objects on our campus. It may have low confidence in some of its initial classifications. Given a limited human query budget and a motion budget, the robot then needs to decide which object locations it should visit to sense better, and which object images it should ask a remote human operator to classify, in order to improve its overall classification performance. By using the correlation coefficient of the feature vectors from a trained DCNN classifier, the robot can robustly capture image similarities (e.g., objects 2 and 3 belong to the same class), which has a significant implication for its field decision-making and joint labeling, as we show in this chapter. Readers are referred to the color pdf to better view the images in this chapter.

based joint labeling framework, where the similarity information is utilized to reduce the classification uncertainty. For instance, in Fig. 5.1, there are two images (2 and 3) of the same class (person) but in different poses. The robot’s vision initially misclassifies them to a truck and a car. However, if the robot is aware that they belong to the same class (while it cannot properly classify them), then it can correctly classify both to persons using our MRF joint labeling framework. This has significant implications for the robot’s field decision-making, such as path planning and human query, as well. Given a number of allowed queries to ask for human help and a motion budget, we finally show how the robot can optimize its motion and human query decisions in order to improve its sensing and scene understanding, by utilizing the inferred visual similarity information.

The rest of the chapter is organized as follows. In Sec. 5.1, we introduce our object

similarity metric and confirm its reliability based on extensive studies. We further propose a Correlation-based Markov Random Field (CoMRF) for joint labeling, based on our inferred object similarity. In Sec. 5.2, we develop our query-motion co-optimization algorithm using our CoMRF. In Sec. 5.3, 5.4, and 5.5, we evaluate our proposed approach for joint labeling, query selection, and path planning, on a large COCO-based test set, a large-scale drone imagery dataset, and a large indoor scene dataset, respectively. In Sec. 5.6, we further demonstrate the efficacy of our proposed algorithm with several robotic experiments. We finally discuss a few more aspects of our methodology in Sec. 5.7.

5.1 Correlation-based Markov Random Field

In this section, we first establish that the correlation coefficient between two feature vectors, from a DCNN-based classifier, provides a reliable metric for characterizing the probability that the two images belong to the same class. We then show how to build a correlation-based Markov Random Field (CoMRF) for joint object labeling, which captures and utilizes our proposed pairwise probabilistic object similarity metric.

5.1.1 Image Similarity and Feature Correlation

Consider a pair of images, each containing an object of interest. The images have passed through a trained DCNN classifier, which automatically provides a feature vector for each image. In contrast to hand-crafted features, this automated feature vector contains robust information on the essence of an object, which allows for capturing image similarity reliably, without additional training, as we show in this section.

To test our hypothesis about this image similarity metric, we construct a large image classification dataset, which contains 39 object classes, including a variety of daily objects (e.g., person, car). There are 76,505 images in total, collected from the COCO

dataset [76] and ImageNet [120]. Most of the images are obtained from the COCO detection dataset by extracting object image patches based on the provided bounding box annotations, in order to better represent what the robot would see in real-world classification tasks (e.g., images that can be small, have occlusion/clutter, and have non-ideal lighting/contrast). We divide this dataset into 38,555 training images, 19,350 validation images, and 18,600 test images.¹ Utilizing this dataset, we have trained DCNN classifiers using the following three commonly-used state-of-the-art architectures: AlexNet [48], MobileNet-v2 [50], and ResNet-18 [51], with their respective accuracies over the validation set as follows: 0.800, 0.873, and 0.873. We refer to these as the base classifiers in the chapter. By utilizing these three different network architectures, we will establish the generalizability of our metric to different networks. Each trained DCNN classifier then automatically provides a feature vector for an input image from the layer prior to the final output layer. For instance, in the case of AlexNet, this feature vector is provided by the activations on the 7th layer, which is a fully-connected layer prior to the final output layer. The test set is reserved for evaluating our proposed methodology in Sec. 5.3.

Remark 5.1 *We note that we could have used an existing trained classifier to test our image similarity metric. However, we chose to train a classifier in order to train on classes more relevant to the types of objects that an unmanned vehicle may see on a campus (e.g., people, bikes, cars). The unmanned vehicle will then use this classifier for our real-world experimental tests in Sec. 5.6. We thus emphasize that the aforementioned training is simply to train a classifier and our proposed image similarity metric is freely available from any DCNN classifier.*

During the training of a DCNN classifier, the network learns relevant features to feed to the output layer, which is a linear operator. As such, the DCNN is trained to derive

¹Detailed descriptions of this dataset can be found in Appendix B.1.

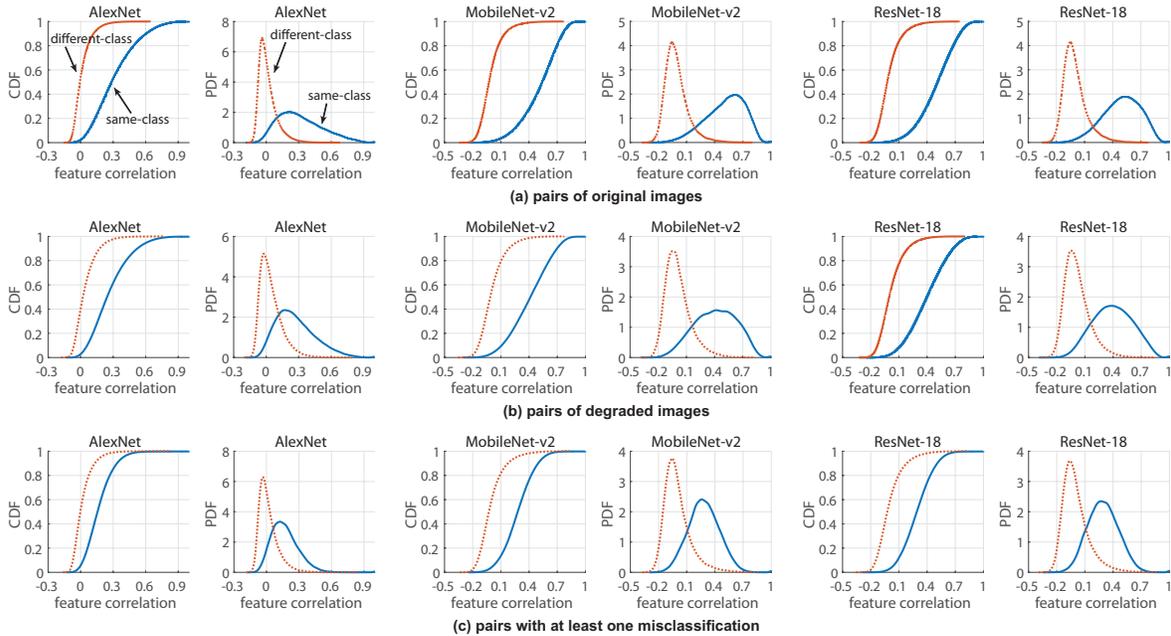


Figure 5.2: (a) In each CDF-PDF figure pair of a DCNN classifier (e.g., the two leftmost figures of AlexNet), the left figure shows the CDFs of the feature correlation of same-class (blue solid) and different-class (red dashed) pairs, and the right figure shows the PDFs of the feature correlation of same-class and different-class pairs. (b) CDFs and PDFs of the feature correlation after degrading the images by low resolution and low illumination. (c) CDFs and PDFs of the feature correlation with at least one misclassification in each image pair.

features to linearly separate images of different classes. Motivated by this generic design of DCNN classifiers, we utilize the Pearson linear correlation coefficient to measure the similarity between a pair of images. More specifically, given a feature vector for each image in a pair of images, we compute the Pearson linear correlation coefficient of the feature vectors of the two images. We refer to this metric as the *feature correlation*. We next empirically analyze its distribution. Fig. 5.2 (a) shows the Cumulative Distribution Functions (CDFs) and Probability Density Functions (PDFs) of the feature correlation of 180,000 random image pairs from the validation set, based on the features provided by the trained AlexNet, MobileNet-v2, and ResNet-18, respectively. More specifically, in each CDF-PDF pair (e.g., the two leftmost figures of AlexNet in Fig. 5.2 (a)), the

blue solid curve of the left figure shows the CDF of the feature correlation of a pair of object images, given these two objects belong to the same class, which we denote as a *same-class pair*. The red dashed curve then shows the CDF of the feature correlation of a pair of objects belonging to different classes, which we denote as a *different-class pair*. It can be seen that, for each DCNN classifier, there is a considerable difference between these two distributions. For a different-class pair, the feature correlation is more likely to be small, while for a same-class pair, there is a higher chance of a high correlation. Similarly, the PDF curves show that the distributions of the correlation coefficient are well separated for same-class and different-class pairs. This further motivates utilizing feature correlation to deduce whether two objects belong to the same class. To do so, we need a threshold, above which to declare two objects in the same class and below which to declare otherwise. We choose a threshold such that the probability of false correlation (different-class declared as same-class) is very small. For instance, for AlexNet, we can see that the probability of a different-class pair having a correlation above 0.3 is less than 0.010, while 47.3% of the same-class pairs have a correlation above 0.3. We then use 0.3 as our threshold for AlexNet in our experiments. Similarly, we select thresholds for the MobileNet-v2 and ResNet-18 classifiers such that the probability of false correlation is very small, while a large percentage of same-class pairs still have a correlation above the threshold. This allows us to capture many of the same-class objects, while ensuring a very small probability of mistaking a different-class pair for a same-class pair. The respective thresholds for the three DCNN classifiers are summarized in Table 5.1, along with the probability of false correlation and the probability of same-class pairs having a correlation above the threshold. When two images have a high feature correlation, there is a high probability that they belong to the same class. Therefore, when designing our CoMRF in Sec. 5.1.2, every two images with a feature correlation above the threshold are connected by an edge and are more likely to have the same label in the joint labeling

Base classifier	Correlation threshold	Prob. above threshold	
		<i>different-class</i>	<i>same-class</i>
AlexNet	0.3	0.009	0.473
MobileNet-v2	0.4	0.009	0.720
ResNet-18	0.4	0.009	0.693

Table 5.1: Correlation thresholds for the three DCNN classifiers. The third and fourth columns show the probability of false correlation (different-class declared as same-class) and the probability of a same-class pair having a correlation above the threshold, respectively.

process.

Robustness to Visual Differences: As two instances of the same object class can be very different visually, it is important that our image similarity metric declares many of such instances to be in the same class. We extensively study our dataset from this angle, in order to ensure that it contains a variety of poses/views for each object class. The PDF/CDF plots of Fig. 5.2 (a), for instance, are obtained from a general dataset (the validation part of our constructed dataset) with visually-diverse instances of each object class. Fig. 5.3 shows 10 random samples of bicycle for instance. Using the AlexNet classifier and based on the threshold of 0.3, a link is drawn between two objects if their feature correlation is above 0.3. As can be seen, despite the drastic visual differences, the resulting graph is dense (55.56% of all possible pairwise links are captured), indicating that the feature correlation robustly captures same-class objects.

Robustness to Image Degradation: In practice, robot sensing may suffer from various degradations, such as low resolution and low illumination. As our approach relies on the feature correlation of image pairs, it is important to understand its robustness to image degradation. In other words, the difference between the feature correlation distributions of same-class and different-class pairs should still be large enough. To evaluate this, we have corrupted the validation set images by randomly reducing resolution and illumination. Fig. 5.2 (b) shows the CDF/PDF of the feature correlation for the corrupted image

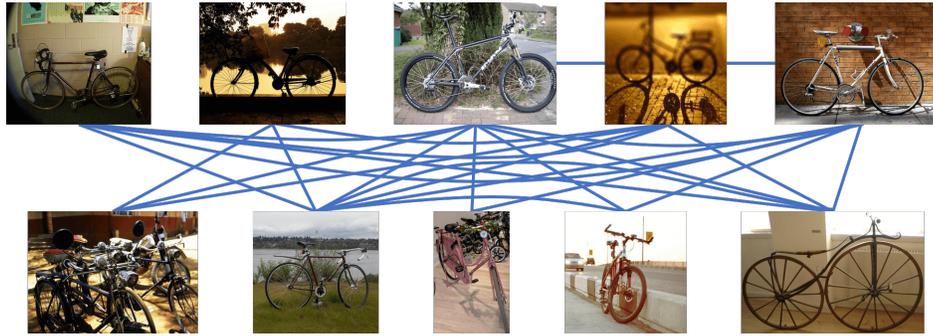


Figure 5.3: Robustness of our metric to visual variations – visualization of the feature correlation among 10 random bicycle images. Two images are connected if their AlexNet-based feature correlation is above the threshold. The number of edges in this graph is 55.56% of that of a 10-node complete graph, indicating that many of the pairs are correctly declared as same-class despite drastic visual differences.

dataset and the three network architectures. As can be seen, although the difference between the two distributions has become smaller, they are still robustly different.

Robustness to Misclassification: The feature correlation can identify same-class objects, even when the classifier misclassifies them. This is important as when there are misclassified images, the feature correlation should still robustly capture the similarity, which can then be utilized to correct the misclassifications. The initial image pool used to plot Fig. 5.2 (a) includes several image pairs with at least one misclassified image. In order to more explicitly show the robustness of the feature correlation to misclassification, Fig. 5.2 (c) shows the feature correlation distributions of same-class and different-class pairs when at least one of the images in a pair is misclassified. It can be seen that, for each DCNN classifier, using the corresponding threshold, the feature correlation still captures many same-class pairs with a small false correlation probability (less than 0.018 for all three DCNN classifiers) in this challenging setting. Fig. 5.1 shows an example of this where, using the AlexNet classifier, the two human images (2 and 3) are initially misclassified as a truck and a car. Utilizing the feature correlation, the robot can infer that they are highly likely to be in the same class, which allows it to jointly label them

correctly, using the method we propose in Sec. 5.1.2.²

5.1.2 Correlation-based Markov Random Field (CoMRF)

So far, we have established that feature correlation provides a reliable metric for image similarity. We next show how it can be used for joint labeling. Suppose that we have N images, each containing an object-of-interest (defined as an object belonging to the set of classes with which the classifier was trained). We next construct our CoMRF based on the pairwise feature correlations.³

In our MRF, each object image is represented by a node and two nodes are connected by an edge if their pairwise feature correlation is above a certain threshold s_T (e.g., the threshold for the AlexNet classifier is 0.3, as discussed in Sec. 5.1.1). The overall potential function of the CoMRF is then given as follows:

$$P(x_1, \dots, x_N) = \prod_{i=1}^N \phi_i(x_i) \prod_{(i,j) \in \tilde{L}} \psi(x_i, x_j), \quad (5.1)$$

where $x_i \in \{1, \dots, N_c\}$ is the label variable of the i^{th} image with N_c denoting the total number of object classes, $\phi_i(x_i)$ is the node potential function, $\tilde{L} = \{(i, j) | \tilde{s}_{i,j} \geq \tilde{s}_{\text{Th}}\}$ is the set of pairs with feature correlation $\tilde{s}_{i,j}$ above \tilde{s}_{Th} , and $\psi(x_i, x_j)$ is the edge potential function.

Node Potential: For each node i , $\phi_i(x_i) = p_{X_i}(x_i)$, where $p_{X_i}(x_i)$ is the probability distribution over the classes from the classifier's output and $x_i \in \{1, \dots, N_c\}$ is the label variable.

Edge Potential: Given that there is an edge between nodes i and j , we denote the

²Even when the robot's initial classifications are correct, it may have low confidence on some of the images. In such cases, the robot can still benefit tremendously from our similarity metric, which can be utilized to reduce the uncertainty in the robot's classification decisions using our CoMRF method of Sec. 5.1.2.

³Readers are referred to the MRF literature for more details on the terminology (e.g., [121, 122]).

probability that these two nodes have the same label as $p_{\text{same}} = p(x_i = x_j | \tilde{s}_{i,j} \geq \tilde{s}_{\text{Th}})$, where $\tilde{s}_{i,j}$ is the feature correlation between nodes i and j . If $x_i = x_j$, we assume that it is equally probable for nodes i and j to belong to any one of the N_c classes. Similarly, if $x_i \neq x_j$, then we assume that it is equally probable for nodes i and j to take any pairwise combination of the N_c classes. The edge potential function is then constructed as follows: $\psi(x_i, x_j) = p_{\text{same}}/N_c$, if $x_i = x_j = k$, $\forall k \in \{1, \dots, N_c\}$, and $\psi(x_i, x_j) = (1 - p_{\text{same}})/(N_c^2 - N_c)$, if $x_i \neq x_j$, $\forall x_i, x_j \in \{1, \dots, N_c\}$.

In a specific joint labeling task instance, given that the objects present belong to $N_p \leq N_c$ classes (subset out of the total N_c classes seen in training) and the threshold is \tilde{s}_{Th} , p_{same} can be written as follows using the Bayes rule:

$$\begin{aligned} p_{\text{same}} &= p(x_i = x_j | \tilde{s}_{i,j} \geq \tilde{s}_{\text{Th}}) \\ &= \frac{p(\tilde{s}_{i,j} \geq \tilde{s}_{\text{Th}} | x_i = x_j)p(x_i = x_j)}{p(\tilde{s}_{i,j} \geq \tilde{s}_{\text{Th}} | x_i = x_j)p(x_i = x_j) + p(\tilde{s}_{i,j} \geq \tilde{s}_{\text{Th}} | x_i \neq x_j)p(x_i \neq x_j)}, \end{aligned}$$

where $p(x_i = x_j) = N_p/N_p^2$ and $p(x_i \neq x_j) = N_p(N_p - 1)/N_p^2$ are the prior probabilities that two nodes belong to the same class and different classes, respectively.

Since this is dependent on N_p , which the robot does not know during the operation, we average p_{same} over the distribution of N_p . For instance, in practice, we do not expect N_p to be larger than 10 in a task instance. Thus, we assume that N_p is uniform over $\{1, \dots, 10\}$, and numerically evaluate $\mathbb{E}[p_{\text{same}}]$ to be 0.928, 0.904, and 0.878 for AlexNet, MobileNet-v2, and ResNet-18, respectively. For the rest of the chapter, for each DCNN classifier, we then set $p_{\text{same}} = \mathbb{E}[p_{\text{same}}]$ in our CoMRF implementation. In order to compute the posterior distribution of the nodes, we use Loopy Belief Propagation (LBP), which is an approximate inference algorithm [121]. The final estimated label for a node is then given by $\hat{x}_i = \operatorname{argmax} \tilde{p}_{X_i}(x_i)$, where \tilde{p}_{X_i} is the posterior marginal distribution of node i over

the N_c classes, after running LBP on CoMRF.⁴

5.2 Optimization of Querying and Path Planning

Consider the case where the robot is tasked with object classification in an area. The robot does an initial classification based on visual sensing and the state-of-the-art DCNN classifier. However, its classification confidence may not be high for several objects. The robot is given a query budget to ask for human help and/or a motion budget to move to some of the object locations to sense better. In this section, we propose our methodology for co-optimizing query selection and path planning, based on our proposed CoMRF. More specifically, when the robot visits a site to better sense an object, or queries the human, it can obtain the correct label of the corresponding object with a high probability. Given these new labels, the robot can perform conditional inference over CoMRF and update all the remaining labels utilizing our similarity metric. Thus, the robot’s site sensing and human query directly affect its joint labeling.

Ideally, the robot should select a subset of nodes to query and/or visit such that the posterior joint uncertainty of all the nodes on the MRF is minimized. However, the computational complexity of finding the optimum to this problem is very high, and existing methods either resort to greedy schemes or are limited to chain-structure graphs [69,124]. Therefore, we instead consider a *neighborhood uncertainty measure* that can still capture the feature correlation and object similarity. More specifically, for each node, we propose an uncertainty measure that takes into account both its individual uncertainty and the uncertainty of its neighboring nodes in CoMRF. This is because as the correct label is applied to a node, not only is its own uncertainty eliminated, but the uncertainty of the neighbors will also be reduced. This measure then provides a way to quantify the amount

⁴We use the publicly available MRF library [123] for our implementation.

Algorithm 4: Proposed query selection and path planning co-optimization (CoMRF-Opt)

$$\begin{aligned}
& \max_{\gamma_h, \gamma_s, z, \tilde{u}} (\gamma_h + \gamma_s) \cdot [\tilde{r}, 0]^T \\
& \text{s.t.} \quad (1) \quad \sum_{j \in \{\tilde{A}(i) \cup i\}} \gamma_{h,j} + \gamma_{s,j} \leq 1, \quad \forall i \in \{1, \dots, N\}, \\
& \quad (2) \quad \sum_{i=1}^{N+1} \sum_{j=1}^{N+1} z_{i,j} \tilde{d}_{i,j} \leq \mathcal{E}_{\max}, \\
& \quad (3) \quad \sum_{j=1}^{N+1} z_{i,j} = \sum_{j=1}^{N+1} z_{j,i} = \gamma_{s,i}, \quad \forall i \in \{1, \dots, N+1\}, \\
& \quad (4) \quad \tilde{u}_i - \tilde{u}_j + 1 \leq N \cdot (1 - z_{i,j}), \quad \forall i, j \in \{2, \dots, N+1\}, \\
& \quad (5) \quad \mathbf{1}^T \gamma \leq M, \quad (6) \quad \gamma + \eta \preceq \mathbf{1}, \\
& \quad (7) \quad \gamma_{h,N+1} = 0, \quad \gamma_{s,N+1} = 1, \\
& \quad (8) \quad \gamma_h, \gamma_s \in \{0, 1\}^{N+1}, \quad z \in \{0, 1\}^{(N+1)^2}, \quad \tilde{u} \in [2, N+1]^N
\end{aligned}$$

of (neighborhood) uncertainty reduction, should a node be given the correct label. Our uncertainty measure is then as follows, for each node i ,

$$\tilde{r}_i = (1 - c_i) + g_w \sum_{j \in \tilde{A}(i)} (1 - c_j), \quad \forall i \in \{1, \dots, N\}, \quad (5.2)$$

where $c_i = \max \tilde{p}_{X_i}(x_i)$, with \tilde{p}_{X_i} being the posterior marginal distribution of node i (after LBP), $x_i \in \{1, \dots, N_c\}$ is the label variable. $1 - c_i$ is then a measure of the individual uncertainty of node i . The second term measures the uncertainty of the neighboring nodes of node i , where $\tilde{A}(i)$ denotes the neighbor set of node i . $g_w \geq 0$ weighs the respective importance of the individual and neighborhood uncertainties.⁵

Given this uncertainty measure, we next formulate our query selection and path planning co-optimization, which selects the nodes that maximize uncertainty reduction

⁵We set $w = 5$ in our implementation, based on running AlexNet on the validation set. The weight $g_w = 5$ is then used for all three DCNN classifiers for all the experiments in Sec. 5.3, 5.4, and 5.6

and avoids choosing highly-correlated nodes, under limited query and motion budgets.⁶ We assume that the robot has to return to its initial position after completing the close-up sensings, forming a tour. The optimization formulation is described in Alg. 4, where N is the total number of objects that the robot has initially sensed, $\gamma_h = [\gamma_{h,1}, \dots, \gamma_{h,N+1}]$ denotes the binary decisions of querying the nodes, $\gamma_s = [\gamma_{s,1}, \dots, \gamma_{s,N+1}]$ denotes the binary decisions of visiting the nodes, $\gamma_{h,N+1}$ and $\gamma_{s,N+1}$ are augmented variables for the robot's initial position, $\tilde{r} = [\tilde{r}_1, \dots, \tilde{r}_N]$ is the uncertainty vector from Eq. (5.2), \mathcal{E}_{\max} is the motion budget in terms of total traveled distance, and M is the query budget. $\gamma_{h,i}=1$ indicates that the robot will query object i (with 0 denoting otherwise). $\gamma_{s,i} = 1$ indicates that the robot will visit object i (with 0 denoting otherwise).

In Constraint (1), we impose that for each node i , at most one node can be selected from the set of node i and its neighbors, which prevents the simultaneous selection of highly-correlated nodes. Constraints (2)-(4) are related to the robot's tour planning. Constraint (2) limits the total traveled distance by \mathcal{E}_{\max} , where $\tilde{d}_{i,j}$ is the distance between objects i and j , and $z_{i,j} \in \{0, 1\}$ indicates whether to include edge (i, j) in the tour. Constraint (3) restricts that an object location can only be entered and exited once if it is in the tour ($\gamma_{s,i} = 1$). Constraint (4) is the Miller-Tucker-Zemlin (MTZ) constraint that eliminates sub-tours [125]. Constraint (5) limits the number of queries by M . Constraint (6) prohibits the robot from both querying an object and visiting it. Constraint (7) ensures that the initial robot position is part of the tour. The last set of constraints enforce that all the decision variables (γ_h , γ_s , and z) are binary, and that the MTZ variables are in $[2, N + 1]$.

In the solution of Alg. 4, it is possible that not all the given query and motion budgets are utilized, since we enforce Constraint (1) to avoid selecting highly-similar

⁶In this optimization formulation, the robot is assumed to obtain the correct label if it visits an object. In the experiments of Sec. 5.6.3, the reported performance is based on the actual obtained images after the visits.

nodes. Suppose that Ω_{act} is the set of nodes queried or visited in this solution. If there are any unused queries and/or motion budget, the robot then re-runs a slightly-modified version of Alg. 4, where $w = 0$, Constraint (1) is removed, and the nodes in Ω_{act} are enforced to be queried or visited. In this way, the robot prioritizes the selection of the most important nodes in CoMRF, while ensuring that all the resources are properly utilized. For the rest of the chapter, we refer to our proposed query selection and path planning approach as **CoMRF-Opt**.⁷

5.3 Performance Evaluation on Large COCO-based Test Set

In this section, we evaluate our proposed CoMRF-based co-optimization approach on joint labeling, query selection, and path planning on the large COCO-based test set described in Sec. 5.1.1. We first assume no query or motion budgets for the robot, and evaluate our proposed joint labeling method. We then allow several queries for the robot (no motion) and evaluate our query selection method. Lastly, we incorporate the element of motion and evaluate our query-motion co-optimization approach by simulating the motion.

5.3.1 Joint Labeling

In this part, we assume zero query and motion budgets for the robot, and analyze the joint labeling performance. We compare with the state-of-the-art methods of Cao et al. [64] and Hayder et al. [65], which are similarity-based approaches that use hand-crafted image features to deduce image similarity for joint labeling. We further compare

⁷The optimization in Alg. 4 is a Mixed Integer Linear Program (MILP) and we use Matlab’s MILP solver in Secs. 5.3, 5.4, 5.5, and 5.6.

Base classifier	Independent (base classifier output)	Cao et al. [64]	Hayder et al. [65]	CoMRF (proposed)
AlexNet	0.437	0.438	0.543	0.673
MobileNet-v2	0.708	0.422	0.587	0.900
ResNet-18	0.724	0.422	0.604	0.881
average	0.623	0.427	0.578	0.818

Table 5.2: Performance comparison of joint labeling methods (case of no query or motion). It can be seen that our proposed joint labeling method improves the base classifier by 0.195, on average, in terms of classification accuracy. On the other hand, Cao et al. and Hayder et al. perform similar or worse as compared to the initial classification.

with the benchmark of directly using the base classifier’s output, to which we refer as “independent”.⁸

For each test case, we randomly draw N_p classes from the total $N_c = 39$ classes. For each selected class, we then sample N_I images. In this section, we use $N_p = 2$ and $N_I = 50$, and report the average classification accuracy over 100 random test cases. To make the classification task more challenging, the images are randomly sampled from the set of images whose AlexNet initial classification confidence is below 0.9.

Table 5.2 compares the performance of the independent approach, Cao et al., Hayder et al., and our proposed CoMRF-based joint labeling. We can see that our approach considerably outperforms the independent approach when using any of the DCNN classifiers, with an average improvement of 0.195 in terms of classification accuracy. On the other hand, Cao et al. and Hayder et al. provide only slight improvement in the case of AlexNet, and underperform, as compared to the base classifier, in the cases of the other two DCNNs. This is because their hand-crafted features cannot properly capture object similarity, especially when there is a large number of classes with a wide variety of poses and views for each class.

⁸More details of these methods can be found in Appendix B.2.

5.3.2 Query Selection

We next evaluate our proposed query selection approach (using Alg. 4 with a zero motion budget). In this evaluation, we compare with Wang et al. [70], which is a state-of-the-art Bayesian approach. Since this approach is for selecting queries on a given MRF and not on image similarity, we use our CoMRF, but then apply their query strategy instead of our proposed Alg. 4. We also include a benchmark that greedily selects the nodes with the highest individual uncertainties (from the base classifier), without utilizing any correlation, to which we refer as “independent”.⁹

The test cases used in this part are the same as in Sec. 5.3.1, and the reported classification accuracy is averaged over the 100 test cases. Fig. 5.4 compares the performance of our CoMRF-Opt (red solid) with the independent approach (blue dashed) and Wang et al. (green dashed), when using the three base classifiers, respectively. As can be seen, our proposed approach considerably outperforms both of them. For instance, when using AlexNet as the base classifier, given a budget of 40 queries, CoMRF-Opt achieves a classification accuracy of 0.957, as compared to 0.878 of Wang et al. and 0.704 of the independent approach. As for the other two DCNNs, although CoMRF already achieves a high initial classification accuracy of ~ 0.900 , CoMRF-Opt is still able to further improve it to ~ 0.990 with 20 queries, significantly outperforming Wang et al. and the independent approach. Furthermore, our proposed approach enables significant resource savings. For instance, for the case of AlexNet, in order to achieve an average classification accuracy of 0.900, our proposed method requires 25 queries, while Wang et al. and the independent benchmark require 50 and 80 queries, respectively, as shown in Fig. 5.4 (a). This is equivalent to respective reductions of 50% and 68.75% in terms of communication resources.

⁹More details of these methods can be found in Appendix B.2.

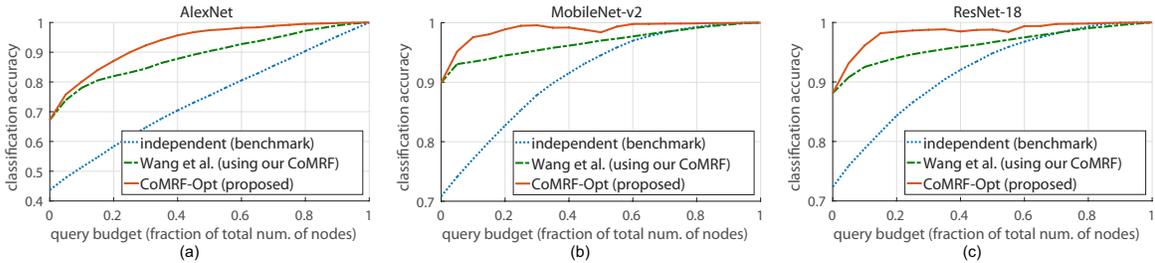


Figure 5.4: Joint labeling and query selection performance (no motion) on the CO-CO-based test set. The query budget is given as a fraction of the total number of nodes.

Since Cao et al. and Hayder et al. only perform comparable to or worse than the base classifier, and Wang et al. only provides near-linear improvement w.r.t. the number of given queries, we will not include them for comparison in the rest of the chapter.

5.3.3 Query Selection and Path Planning

In this section, we take motion into account and evaluate our CoMRF-based query-motion co-optimization approach on our COCO-based test set by running the robot in a simulated motion environment. More specifically, for each test case, we randomly draw $N_p = 2$ classes from the total $N_c = 39$ classes and for each selected class, we randomly sample $N_I = 10$ images from the set of images whose AlexNet initial classification confidence is below 0.9. These 20 images/objects are then randomly placed in a $10\text{ m} \times 10\text{ m}$ simulation environment. In each test case, given a query budget and a motion budget, the robot needs to decide for which object images it should query the remote human operator and which object locations it should visit to sense better. We give the robot a query budget equal to 5% of the total number of nodes and a motion budget ranging from 0 m to 20 m. We compare our proposed CoMRF-Opt (Alg. 4) with the independent approach that does not take into account any correlation.¹⁰ We report the average

¹⁰For the independent approach, the path planning part is conducted by running a modified version of Alg. 4, where r in the objective function is replaced by a vector of the individual uncertainties, given by the base classifier, and Constraint (1) is removed.

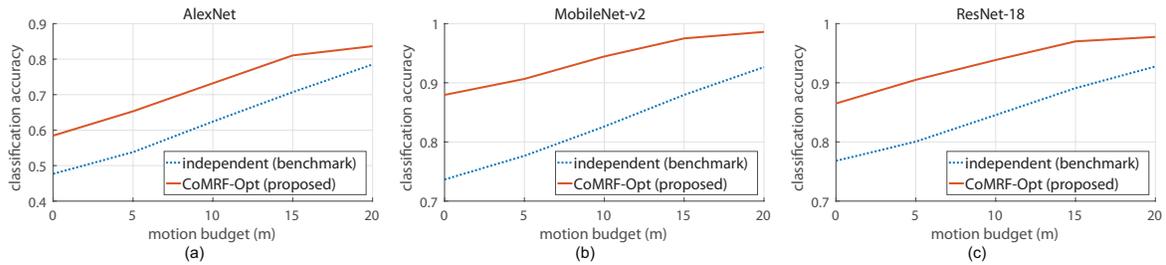


Figure 5.5: Query selection and path planning performance on the COCO-based test set. The robot is given a query budget equal to 5% of the total number of nodes and the motion budget ranges from 0 m to 20 m.

classification accuracy over 100 random test cases.

Fig. 5.5 shows the classification performance of CoMRF-Opt (red solid) and that of the independent approach (blue dashed), for the cases of the three DCNN base classifiers, respectively. It can be seen that CoMRF-Opt significantly outperforms the independent approach for any given motion budget and for all three DCNNs. For instance, in the case of MobileNet-v2, given a motion budget of 10 m, CoMRF-Opt achieves a classification accuracy of 0.945, which is considerably higher than that of the independent benchmark (0.826). For a larger query budget, our proposed approach has a similar performance improvement over the benchmark. For instance, given a query budget of 15% and a motion budget of 10 m, CoMRF-Opt has a classification accuracy of 0.978 and the benchmark’s accuracy is 0.880, when using MobileNet-v2. As the amount of given resources increases considerably, however, both approaches’ performance will approach 1 eventually, as expected.

Overall, these results confirm that our proposed feature correlation metric robustly captures image similarity and our CoMRF-based approach improves classification performance considerably. For the joint labeling task, the accuracy of our proposed approach is significantly higher than those of the state of the art. By comparing with Wang et al., we can see that our proposed CoMRF-based query strategy outperforms one of the best existing approaches. By including motion in the experiments, we further validate

our proposed CoMRF-based query-motion co-optimization methodology.

5.4 Performance Evaluation on a Large-Scale Drone Imagery Dataset

In this section, we evaluate the performance of our proposed CoMRF-based joint labeling and query-motion co-optimization methodology on a large drone imagery dataset. We use the publicly-available VisDrone dataset [77],¹¹ which is a challenging large-scale dataset with images captured by drone-mounted cameras that cover various cities, environments, object classes, and object densities. As images taken by drones have very different views as compared to images taken on the ground (e.g., COCO images), evaluating performance on the VisDrone dataset will further verify the robustness and generalizability of our proposed feature correlation-based similarity metric and the CoMRF-based co-optimization approach. While using the VisDrone data, we consider a subset of their object classes (7 out of 10) that overlaps with the classes of our training set described in Sec. 5.1.1, which includes person, pedestrian, car, bus, truck, motorcycle, and bicycle. The two classes of person and pedestrian are treated as one class in our evaluation. The object image patches are obtained based on the provided bounding box annotations.

In the following parts of this section, we first evaluate the performance of joint labeling and query selection by assuming a zero motion budget. We then allow a non-zero motion budget and evaluate our proposed query-motion co-optimization approach in a simulation environment.

¹¹The dataset is publicly available from <http://aiskyeye.com/>.

5.4.1 Joint Labeling and Query Selection

In the joint labeling and query selection evaluation, we consider two realistic drone visual sensing scenarios. In the first scenario, in each test case, the images are taken from the same flight. This captures a real-world scenario where the drone utilizes not only the correlation among objects within the same image, but also that among objects from different images taken during the same flight, in order to improve its onboard DCNN classifier’s accuracy. We refer to this scenario as the *same-flight scenario*. Secondly, we consider a scenario where in each test case, the images are randomly drawn from the entire dataset. This captures another practical scenario where there are multiple drones performing visual tasks in different environments, while there is a central agent (e.g., a leader drone or a ground robot) that receives images from these drones and performs joint labeling. We refer to this scenario as the *multi-flight scenario*. In both scenarios, each test case contains at least 100 objects from images from either the same flight or multiple random flights, and we report the average classification accuracy over 100 random test cases for each scenario. Note that, unlike in the previous section, the number of classes present (N_p) and the number of images per class (N_I) are not controlled here. In other words, there can be any number of at least 2 object classes present and the number of images for each class can take any value in each test case.

Fig. 5.6 shows the performance of our proposed joint labeling and query selection approach, as compared to the benchmark, in the same-flight scenario. It can be seen that our proposed approach significantly outperforms the benchmark in terms of classification accuracy, when using any of the three DCNN base classifiers. For instance, when using AlexNet, given a query budget equal to 30% of the number of nodes, CoMRF-Opt achieves an accuracy of 0.854, which is 0.424 higher than that of the base classifier (0.430). Similar large improvements can be seen for the cases of MobileNet-v2 and ResNet-18.

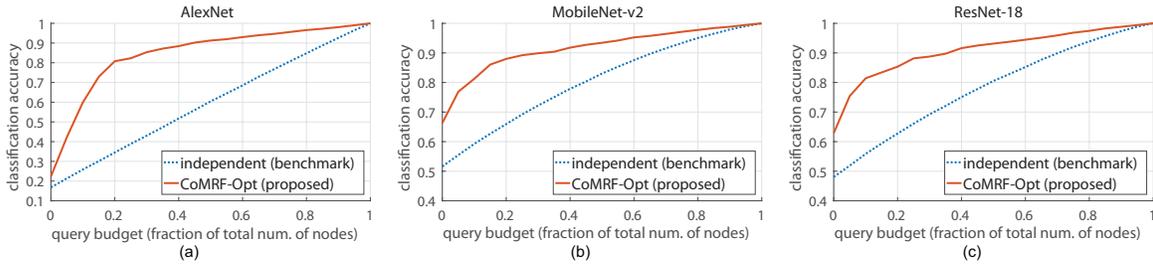


Figure 5.6: Joint labeling and query selection performance (no motion) on the Vis-Drone dataset, for the same-flight scenario. The query budget is given as a fraction of the total number of nodes.

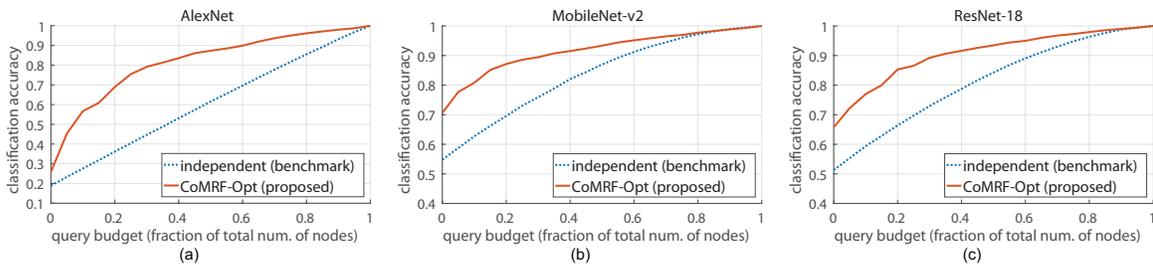


Figure 5.7: Joint labeling and query selection performance (no motion) on the Vis-Drone dataset, for the multi-flight scenario. The query budget is given as a fraction of the total number of nodes.

Fig. 5.7 compares our proposed CoMRF-based approach with the independent benchmark, in the multi-flight scenario. Similarly, it can be seen that our proposed approach significantly outperforms the independent approach, for all three DCNNs. For instance, in the case of AlexNet, given a query budget equal to 30% of the number of nodes, CoMRF-Opt achieves an accuracy of 0.792, while the benchmark has an accuracy of 0.447, which is 0.345 lower. It can also be observed that the performance improvement provided by CoMRF-Opt is slightly less in the multi-flight scenario, as compared to that in the same-flight scenario, since there could be less correlation across images taken from different flights, as expected.

Next, we study how the classification performance varies w.r.t. the number of classes present (N_p) in a test case, in order to understand the effect of N_p on our proposed approach. More specifically, based on the 100 same-flight test cases, we calculate the

N_p	Ave. N_I	AlexNet		MobileNet-v2		ResNet-18	
		Ind.	CoMRF-Opt	Ind.	CoMRF-Opt	Ind.	CoMRF-Opt
≤ 3	41	0.593	0.862	0.804	0.916	0.777	0.909
4	27	0.602	0.844	0.815	0.917	0.796	0.909
5	22	0.585	0.846	0.797	0.921	0.770	0.918
6	18	0.617	0.798	0.799	0.887	0.787	0.862

Table 5.3: Average classification performance given different number of classes present (N_p) in a test case. The first column shows the number of classes present in a test case. The second column shows the average number of objects per class in a test case for different N_p values. The remaining columns show the average classification accuracy of the benchmark of making independent decisions and our proposed CoMRF-based approach, given different N_p values, for the base classifiers of AlexNet, MobileNet-v2, and ResNet-18, respectively.

average classification accuracy for each N_p , where the accuracy is averaged over the test cases with the same N_p and the query budget ranging from 0 to 1. The results are shown in Table 5.3. The first column shows the different N_p values, where $N_p = 2$ and $N_p = 3$ are grouped together to allow for at least 10 cases for averaging. The second column shows the average number of objects per class (N_I) in a test case, for each N_p . The remaining entries show the average classification accuracies of the benchmark of making independent decisions and our CoMRF-based approach. It can be seen that for each base classifier, the benchmark performs similarly across different N_p values, which is as expected as it classifies each object image individually and is thus not affected by N_p . As for our approach, for each base classifier, we can see that the average classification accuracy is also similar across different N_p values. The accuracy is slightly lower when $N_p = 6$, as the average N_I (number of objects per class) is smaller in this case, which means that there is less underlying correlation to be exploited. However, as we can see, our CoMRF-based approach still significantly outperforms the benchmark in this case. These results verify that our proposed approach performs consistently across different N_p values.

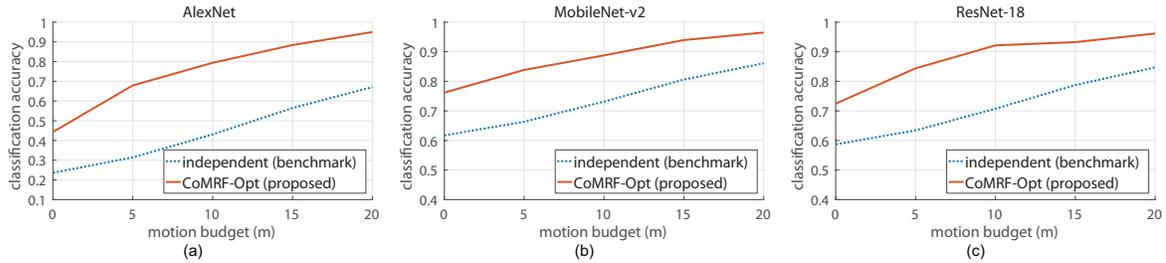


Figure 5.8: Query selection and path planning performance on the VisDrone dataset. The robot is given a query budget equal to 5% of the total number of nodes and the motion budget ranges from 0 m to 20 m.

5.4.2 Query Selection and Path Planning

In this part, we provide a non-zero motion budget to the robot and evaluate our proposed query-motion co-optimization approach. In each test case, an image (with 15 to 25 objects) is randomly drawn from the VisDrone dataset and the objects in this image are randomly placed in a $10\text{ m} \times 10\text{ m}$ simulation environment. By running our proposed algorithm in this setting, we capture a realistic scenario where a drone has acquired an image of the field, and needs to plan its next motion steps to better view the objects and/or select some of the object images to query the remote human operator. We average the performance over 100 random test cases.

Fig. 5.8 shows the performance of our proposed CoMRF-Opt query-motion co-optimization and that of the independent approach, for the three DCNN base classifiers, respectively. It can be seen that overall, CoMRF-Opt significantly outperforms the independent approach. For instance, when using ResNet-18, given a motion budget of 10 m, CoMRF-Opt achieves a classification accuracy of 0.921, as compared to an accuracy of 0.707 by the independent benchmark.

Overall, these results confirm the efficacy of our proposed CoMRF-based joint labeling and query-motion co-optimization approach, showing that CoMRF-Opt can achieve significantly higher classification accuracies as compared to the benchmark of making independent decisions. Furthermore, the visually-challenging VisDrone-based evaluation

also demonstrates the robustness and generalizability of our proposed feature correlation and co-optimization approach.

5.5 Performance Evaluation on a Large Indoor Scene Dataset

We further evaluate the performance of our proposed CoMRF-based joint labeling and co-optimization methodology on the popular indoor scene dataset of NYU-v2 [78].¹² This large indoor dataset contains a variety of scenes (e.g., kitchens, offices) and a large number of various objects. As indoor objects can be challenging to recognize, e.g., due to occlusion and clutter, this evaluation will further verify the robustness of our proposed approach. In the evaluation, we consider a set of 19 object classes that are present in both the NYU-v2 dataset and our training set in Sec. 5.1.1. The object image patches are obtained based on the provided annotations.

In the evaluation, we use the objects of the same scene for each test case, which captures a real-world situation where a robot enters a scene and needs to recognize the objects in the scene. We randomly select 100 test cases (i.e., 100 scenes) from the dataset, each contains at least 10 objects. The reported performance is averaged over the 100 test cases. Next, we present the evaluation on joint labeling, query selection, and query-motion co-optimization for our proposed approach.

5.5.1 Joint Labeling and Query Selection

Fig. 5.9 shows the performance of our proposed joint labeling and query selection approach, as compared to the benchmark of making independent decisions. It can be seen

¹²The dataset is publicly available from https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html.

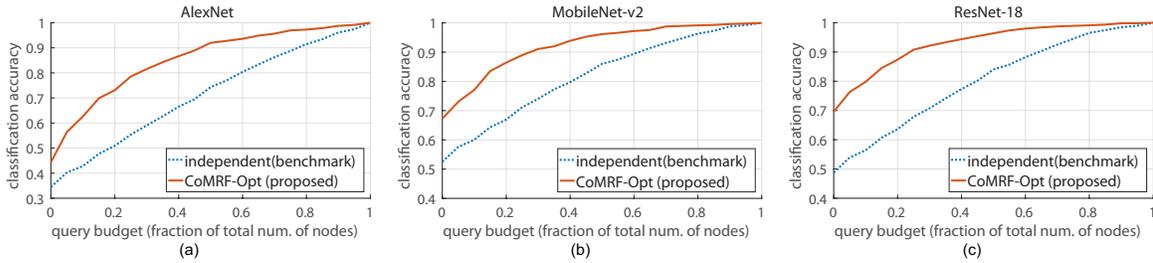


Figure 5.9: Joint labeling and query selection performance (no motion) on the NYU-v2 dataset. The query budget is given as a fraction of the total number of nodes.

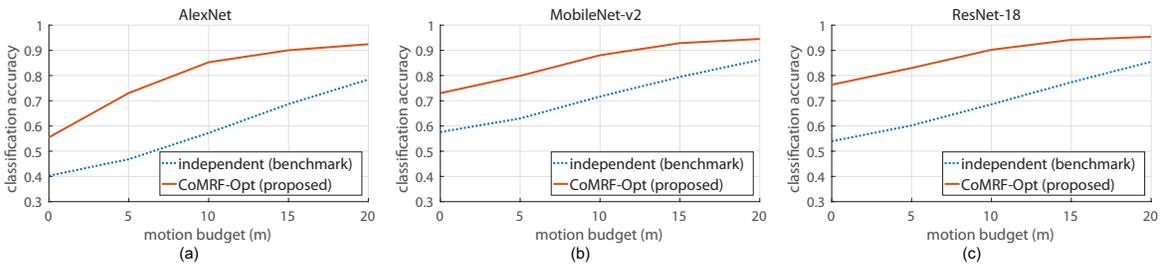


Figure 5.10: Query selection and path planning performance on the NYU-v2 dataset. The robot is given a query budget equal to 5% of the total number of nodes and the motion budget ranges from 0 m to 20 m.

that our proposed approach significantly outperforms the independent approach, when using any of the three DCNN base classifiers. For instance, given a query budget equal to 30% of the number of nodes, CoMRF-Opt achieves an accuracy of 0.815, 0.911, and 0.922, considerably outperforming the benchmark by 0.225, 0.170, and 0.214, respectively, when using the base classifiers of AlexNet, MobileNet-v2, and ResNet-18.

5.5.2 Query Selection and Path Planning

In this part, the robot is given a non-zero motion budget. In each test case, the objects from the same scene are randomly placed in a $10\text{ m} \times 10\text{ m}$ simulation environment. Our evaluation in this section captures a real-world scenario where a robot is tasked with recognizing the objects in an indoor scene, and needs to plan its motion steps to better view the objects and/or select some of the object images to query the remote human

operator.

Fig. 5.10 shows the performance of our proposed CoMRF-Opt query-motion co-optimization approach and that of the independent one, for the three DCNN base classifiers. It can be seen that CoMRF-Opt significantly outperforms the independent approach. For instance, when using AlexNet, given a motion budget of 10 m, CoMRF-Opt achieves a classification accuracy of 0.853, as compared to an accuracy of 0.572 by the independent benchmark.

Overall, these results further confirm the efficacy and robustness of our proposed CoMRF-based joint labeling and query-motion co-optimization approach for indoor scenes, showing that it can achieve a significantly higher classification accuracy as compared to making independent decisions.

5.6 Robotic Experiments

In this section, we evaluate our proposed query selection and path planning approach (CoMRF-Opt) with several robotic experiments on our campus. We also compare its performance with the benchmark of making independent decisions. We have conducted a total of six experiments. In the first three experiments, we evaluate the query selection part, where the robot takes several images around it and is allowed to ask for human help under a query budget (zero motion budget). In the remaining three experiments, we provide the robot with a motion budget, adding path planning into the robot's decision-making. The real-world robotic experiments are conducted with the robot running the AlexNet classifier onboard.

5.6.1 Experiment Overview

In the experiments, we use a Pioneer 3-AT robot, equipped with a ZED camera (with depth sensing) and a laptop. Fig. 5.11 shows the robot's steps during an experiment. The robot first takes several images of its surroundings. For each image, the robot extracts Regions-Of-Interest (ROIs), each of which potentially contains an object-of-interest.¹³ These potential objects are then fed into a background rejector, which is a binary classifier that determines whether an image contains an object-of-interest (e.g., one of the 39 object classes vs. a background wall).¹⁴ Once the background image patches are rejected, the robot passes the object image patches, each containing an object-of-interest, to its onboard DCNN classifier (described in Sec. 5.1.1), which provides the classification output and feature vector for each object image. The robot then constructs the CoMRF for these objects, and optimizes (in real time) the queries and motion using Alg. 4. Given the optimized decisions, the robot performs the corresponding queries and/or further sensing, after which it obtains the labels for the queried/visited nodes and updates the remaining nodes on the CoMRF, using LBP as discussed in Sec. 5.1.2.

5.6.2 Query Selection

In the first three experiments, the robot is tasked with object classification on our campus, and is allowed to query a remote human operator for help. We assume that when the robot queries about an object, it receives the correct label.

¹³The ROI extraction algorithm utilizes the depth and the saliency map of [126]. This algorithm is designed to find coarse ROIs, each of which may contain an object-of-interest, rather than providing tight bounding boxes.

¹⁴Details of the ROI extraction algorithm and the background rejection classifier can be found in Appendices B.3 and B.4.

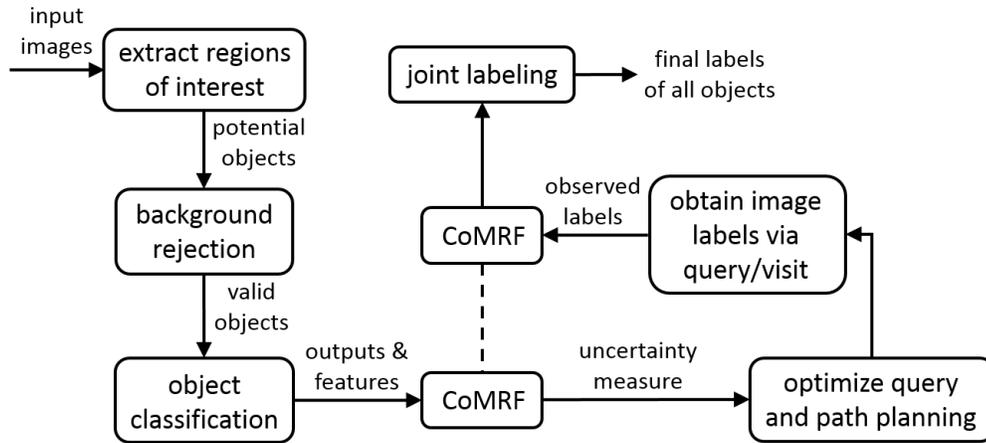


Figure 5.11: The diagram shows the robot’s steps in an experiment using our proposed approach. The dashed line between the two “CoMRF” blocks indicates that they are the same MRF, based on the initial classification and feature extraction. The top CoMRF is then further updated with the image labels obtained from visit or query.

Experiment 1

Fig. 5.12 shows the images taken by the robot in this experiment on our campus. The CoMRF is constructed based on the extracted features and the chosen correlation threshold discussed in Sec. 5.1. In this CoMRF, there is an edge between images 2 and 3, and an edge between images 5 and 6, as can be seen.¹⁵

The robot then performs joint labeling and query selection using this CoMRF, and we compare its performance with the independent approach. Fig. 5.13 shows the two methods’ accuracies with respect to the number of queries. Initially, the classifier mislabels all six images. Although the AlexNet classifier has a good accuracy of 0.800 over the validation set, its performance degrades in real-world scenarios due to low resolution and non-ideal lighting. When there are no allowed queries, CoMRF improves the initial classification by correctly labeling images 2 and 3 to persons (using feature correlation

¹⁵As discussed in Sec. 5.1.1, the robot may not capture exhaustively all the pairwise same-class objects as we set the threshold high to make the probability of false correlation very small. But as these experiments indicate, what it captures can lead to significant performance improvements for free, by utilizing our proposed similarity metric.

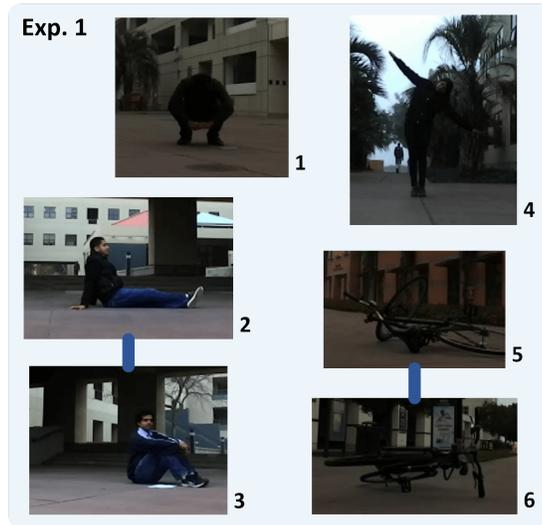


Figure 5.12: Object images obtained by the robot in Exp. 1 on our campus. A line between two images indicates that there is an edge between them in the corresponding CoMRF. See the color pdf to better view all the experiment images.

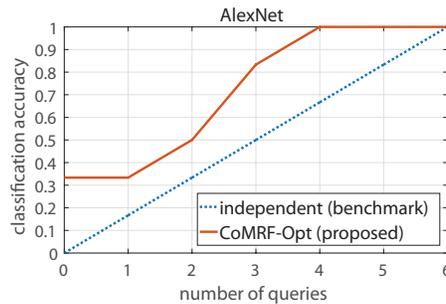


Figure 5.13: Performance (classification accuracy) of CoMRF and the independent approach w.r.t. the number of allowed queries in Exp. 1 (shown in Fig. 5.12).

and joint labeling). Given some queries, we can see that CoMRF-Opt outperforms the independent benchmark significantly. For instance, when given 4 queries, CoMRF-Opt chooses images 1, 3, 4, and 6, and achieves a 1.000 accuracy. On the other hand, the independent approach chooses both nodes 2 and 3 among the 4 queries, which is unnecessary as they are highly similar, and achieves a 0.667 accuracy.

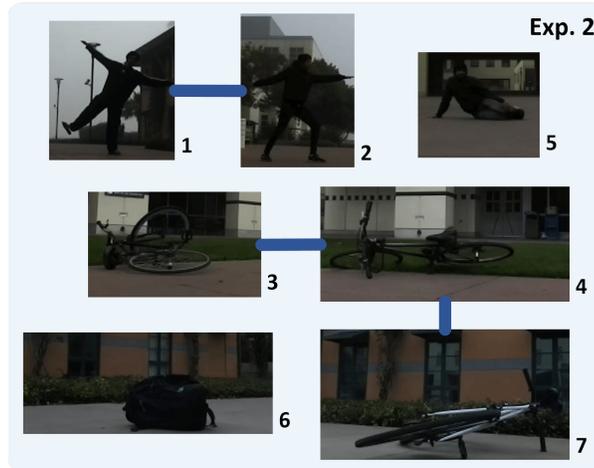


Figure 5.14: Object images obtained by the robot in Exp. 2 on our campus. A line between two images indicates that there is an edge between them in the corresponding CoMRF.

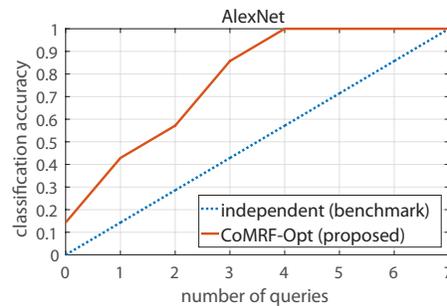


Figure 5.15: Performance (classification accuracy) of CoMRF and the independent approach w.r.t. the number of allowed queries in Exp. 2 (shown in Fig. 5.14).

Experiment 2

Fig. 5.14 shows the images taken by the robot in this campus experiment. In this CoMRF, there is an edge between images 1 and 2, an edge between images 3 and 4, and an edge between images 4 and 7, as can be seen. In particular, the three bicycle images (3, 4, and 7) form a connected component in the graph.

When there are no queries, CoMRF improves the initial classification by correctly labeling node 3 as a bicycle. Initially, nodes 3 and 4 are mislabeled as a potted plant and a bench, respectively, which are the most likely candidates from the classifier for these

two nodes, while bicycle is the second most likely for both nodes. After the similarity between nodes 3 and 4 has been captured in CoMRF, the probability of belonging to the bicycle class increases for both nodes, and for node 3, bicycle becomes the most probable class.

When the robot is given several queries, CoMRF-Opt outperforms the independent approach significantly. For instance, when given 4 queries, CoMRF-Opt chooses nodes 2, 4, 5, and 6, and achieves a 1.000 accuracy. The independent approach, on the other hand, chooses both nodes 1 and 2 among the 4 queries, which are highly correlated, only achieving a 0.571 accuracy.

Experiment 3

Fig. 5.16 shows the images taken by the robot in this campus experiment. In this CoMRF, the persons' images form a connected component, with images 1, 2, and 4 fully interconnected and image 5 connected to image 4. There is also an edge between the bicycle images 6 and 7.

Initially, the based classifier of AlexNet correctly recognizes images 1 and 4, but misclassifies the rest. By applying CoMRF, the robot is then able to classify image 2 as a person, thus improving the classification accuracy without using any queries. When the robot is given a few chances to query, CoMRF-Opt outperforms the independent approach significantly. For instance, when given 3 queries, CoMRF-Opt chooses nodes 3, 5, and 7, and achieves a 100% classification accuracy. On the other hand, the independent approach chooses nodes 4, 6, and 7, among which nodes 6 and 7 are highly correlated, and only achieves a 0.571 accuracy.

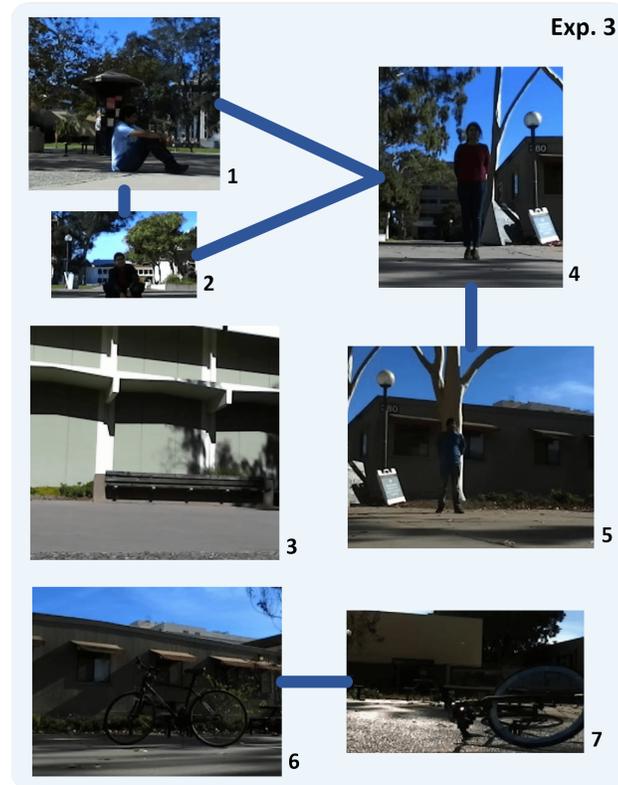


Figure 5.16: Object images obtained by the robot in Exp. 3 on our campus. A line between two images indicates that there is an edge between them in the corresponding CoMRF.

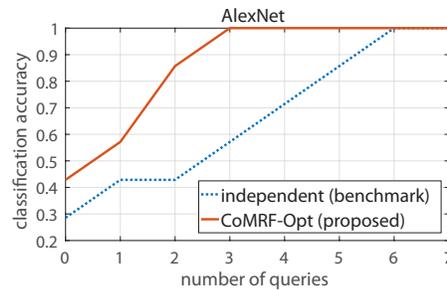


Figure 5.17: Performance (classification accuracy) of CoMRF and the independent approach w.r.t. the number of allowed queries in Exp. 3 (shown in Fig. 5.16).

5.6.3 Query Selection and Path Planning

In this part, we present robotic experiments where the robot is given a non-zero motion budget. When the robot visits an object, it moves towards the object and takes a close-up image. For an object visited by the robot, the label is given by the actual

DCNN classification output based on the close-up image obtained during the visit.¹⁶ The object locations are estimated based on the depth information.

Experiment 4

Fig. 5.18 shows the objects captured by the robot in this experiment, all of which are misclassified initially by the base classifier. As can be seen, CoMRF puts an edge between images 1 and 2. Fig. 5.19 shows the object locations and the robot's initial position. The green dashed line indicates an edge between objects 1 and 2 in the CoMRF. In this experiment, the robot is given no queries and is allowed a total travel distance of 8 m. Fig. 5.19 (a) and (b) show the results of the independent benchmark and our approach, respectively. The independent approach does not take advantage of the feature correlation and chooses to visit the standalone node 3. On the other hand, our proposed CoMRF-based path planning approach of Sec. 5.2 is aware of the correlation and chooses to visit node 2. After node 2 is better sensed and correctly labeled as a person, CoMRF propagates this information to node 1 and then also correctly classifies node 1. Therefore, under the same motion budget, CoMRF-Opt correctly classifies one more node as compared to the benchmark and improves the accuracy by 100% over the benchmark.

Experiment 5

Fig. 5.20 shows the object images captured by the robot in this experiment. In the CoMRF, there is an edge between images 1 and 2, and an edge between images 3 and 4. Fig. 5.21 shows the object locations and the robot's initial position. In this case, the robot is given 1 query and is allowed a total travel distance of 16 m. Fig. 5.21 (a) shows the result of the independent approach. As can be seen, the robot visits nodes 3 and 5, and

¹⁶For the independent approach, we assume that the robot obtains the correct label of the visited node when calculating its performance.

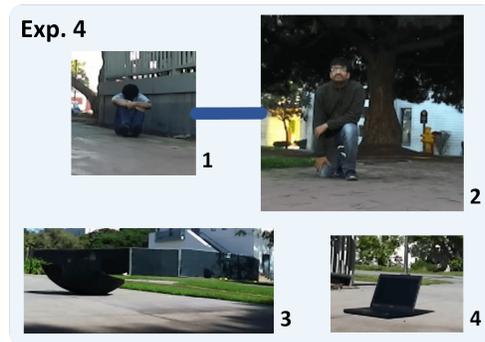


Figure 5.18: Object images obtained by the robot in Exp. 4 on our campus. A line between two images indicates that there is an edge between them in the corresponding CoMRF.

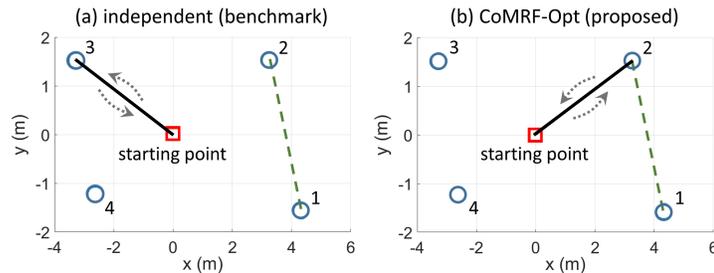


Figure 5.19: Blue circles indicate the object locations and the red square indicates the robot's initial position in Exp. 4 of Fig. 5.18. The green dashed line indicates an edge in the CoMRF. (a)-(b) show the respective path planning results of the independent approach and CoMRF-Opt, where the solid black line indicates the sensing tour.

queries node 4. However, since nodes 3 and 4 are highly correlated, it is unnecessary to obtain labels for both of them. On the other hand, as shown in Fig. 5.21 (b), CoMRF-Opt queries node 1, and visits nodes 4 and 5, which are not highly correlated. Furthermore, due to their influence on their neighboring nodes, CoMRF correctly labels the remaining nodes after the querying and sensing. In this case, the independent approach only obtains a classification accuracy of 0.600, while CoMRF-Opt provides fully correct classifications, significantly improving the accuracy of the benchmark by 0.400.

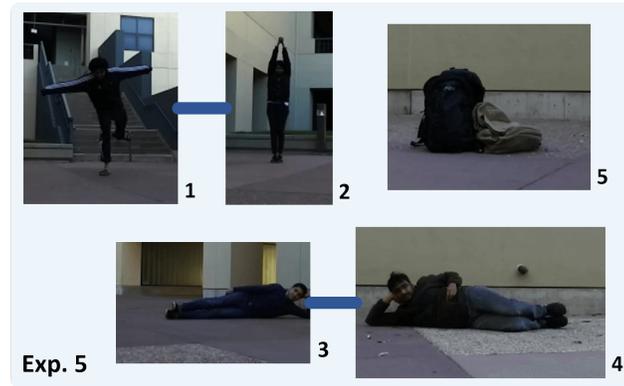


Figure 5.20: Object images obtained by the robot in Exp. 5 on our campus. A line between two images indicates that there is an edge between them in the corresponding CoMRF.

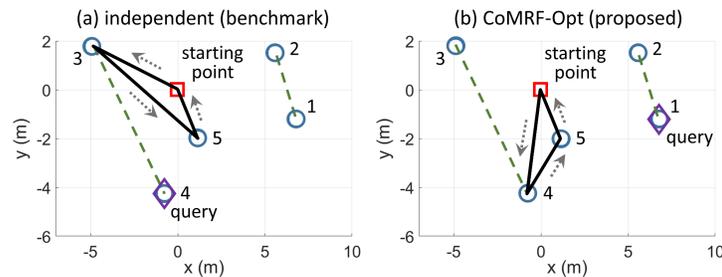


Figure 5.21: Blue circles indicate the object locations and the red square indicates the robot's initial position in Exp. 5 of Fig. 5.20. The green dashed line indicates an edge in the CoMRF. (a)-(b) show the respective path planning results of the independent approach and CoMRF-Opt, where the solid black line indicates the sensing tour and the purple diamond indicates the queried object.

Experiment 6

Fig. 5.22 shows the object images captured by the robot in Exp. 6. In the CoMRF, there is an edge between images 1 and 5, and an edge between images 3 and 4. Fig. 5.23 shows the object locations and the robot's initial position. In this case, the robot is given 1 query and is allowed a total travel distance of 11 m. Fig. 5.23 (a) and (b) shows the result of the independent approach and CoMRF-Opt, respectively. Initially, the robot correctly recognizes images 3 and 4 as benches. By using our proposed approach, the robot visits node 2 and queries node 1, after which it is able to propagate the newly-

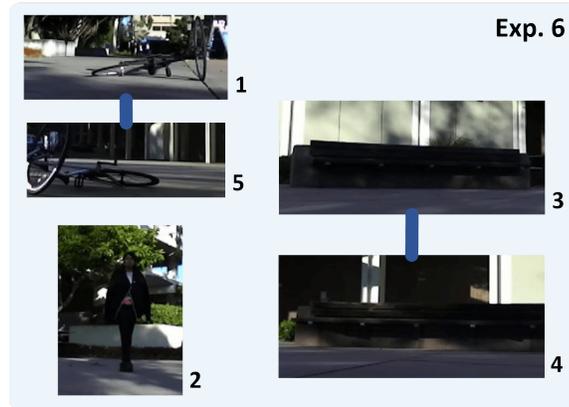


Figure 5.22: Object images obtained by the robot in Exp. 6 on our campus. A line between two images indicates that there is an edge between them in the corresponding CoMRF.

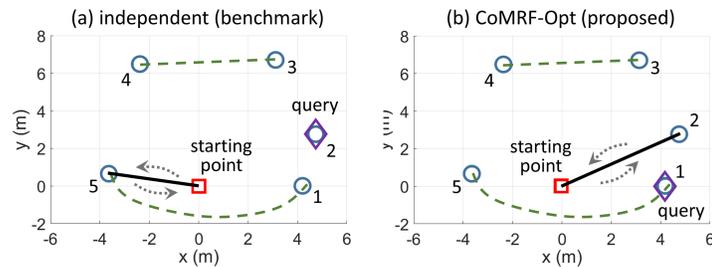


Figure 5.23: Blue circles indicate the object locations and the red square indicates the robot's initial position in Exp. 6 of Fig. 5.22. The green dashed line indicates an edge in the CoMRF. (a)-(b) show the respective path planning results of the independent approach and CoMRF-Opt, where the solid black line indicates the sensing tour and the purple diamond indicates the queried object.

acquired information to node 5 and correctly classifies all the images. On the other hand, the independent approach correctly classifies node 2 and 5 via query/visit, but cannot rectify its initial misclassification of node 1. In this case, the independent approach obtains a final classification accuracy of 0.800, while CoMRF-Opt has a perfect accuracy, considerably outperforming the benchmark by 0.200.

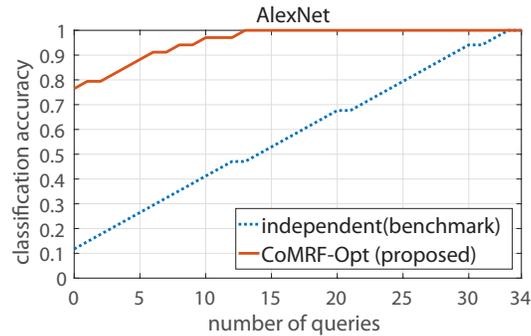


Figure 5.24: Performance (classification accuracy) of CoMRF and the independent approach w.r.t. the number of allowed queries, when jointly classifying all the experimental images (shown in Figs. 5.12, 5.14, 5.16, 5.18, 5.20, and 5.22).

5.6.4 Joint Classification of Objects from Multiple Scenes

In this part, we consider a joint classification scenario where the robot is required to classify the object images that it has acquired during earlier visits of several scenes. This setting captures the case where the robot is not required to produce the classification results on the spot and thus allows the robot to discover a richer correlation structure from a larger image pool. In this evaluation, all the 34 object images obtained from our experiments are jointly classified. Although some of the people appear more than once in this pool, they appear in very different conditions, e.g., clothing, poses, views, lighting, contrast, and scenes. As such, two images of the same person taken at different times/locations cannot be trivially declared to be connected in the graph and their similarity has to be determined by their corresponding DCNN features.

Fig. 5.24 shows the respective classification accuracies of our proposed approach and the independent benchmark in this case. It can be seen that by taking into account the similarity information of all the images, CoMRF is able to greatly improve the classification accuracy without using any queries, from 0.112 to 0.765. By using only 14 queries, CoMRF-Opt achieves a 100% accuracy, while the independent approach requires 33 queries to correctly classify all the images.

Overall, our robotic experiments confirm that the correlation coefficient of two feature vectors, from a DCNN classifier, provides key information on object similarity, and that our proposed CoMRF-based query-motion co-optimization considerably improves the robot’s classification accuracy, as compared to the benchmark of making independent decisions.

5.7 Discussion

In this section, we discuss a few more aspects related to our proposed methodology.

5.7.1 CoMRF on Other DCNNs

We have shown extensive evaluation of our proposed CoMRF-based approach using the AlexNet, MobileNet-v2, and ResNet-18 DCNN architectures. These are the commonly-used state-of-the-art architectures which are typically suitable for mobile computing (e.g., service robots, drones).

There are even deeper architectures that can be used for classification, at the cost of higher computation and memory requirements. Our proposed correlation-based image similarity and joint labeling are also applicable to such larger and deeper networks, such as Inception-v3 [49] and ResNet-101 [51]. To illustrate this, we have trained these two networks using the training set of Sec. 5.1.1. As shown in Fig. 5.25, for these two DCNNs, there is a large separation between the distributions of the feature correlation of different-class and same-class pairs. We then run the same joint labeling evaluation, as in Table 5.2 of Sec. 5.3, using these deeper base classifiers. By using Inception-v3, our proposed approach achieves a classification accuracy of 0.937, as compared to the base classifier’s accuracy of 0.814, and by using ResNet-101, our approach achieves a high accuracy of 0.959, as compared to the initial accuracy of 0.818. This demonstrates that

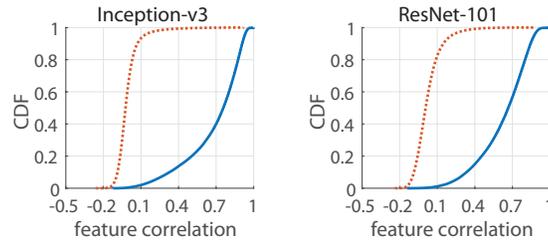


Figure 5.25: Each figure shows the CDFs of the feature correlation of same-class (blue solid) and different-class (red dashed) pairs, based on the Inception-v3 and ResNet-101 classifiers, respectively.

our proposed CoMRF is also able to provide a large accuracy improvement with deeper DCNN classifiers.

Furthermore, the performance improvement provided by our proposed query/motion optimization approach also applies to these deeper DCNNs. For instance, when we run the same-flight query selection evaluation on the VisDrone dataset (as described in Sec. 5.4.1), using these two deeper classifiers, with a query budget equal to 10% of the number of objects, our proposed CoMRF-Opt improves the accuracy of the independent benchmark from 0.636 to 0.793 in the case of Inception-v3 and from 0.689 to 0.844 in the case of ResNet-101.

5.7.2 Computation Time

In this part, we discuss the computation efficiency of CoMRF-Opt. The timing experiments were run in Matlab on a 3.40 GHz Intel Core i7 PC. The reported times are averaged over 100 problem instances. For a test case with 100 object images/nodes, our proposed query selection algorithm (zero motion budget) only takes an average of 0.012 s to produce a solution. As for the query-motion co-optimization (non-zero query and motion), it takes an average of 0.101 s, 1.363 s, 20.217 s, and 101.585 s to solve a problem with 10, 20, 30, and 40 object images, respectively.

In the path planning part of our algorithm, the number of binary variables increases

quadratically with respect to the number of objects, as the robot needs to decide whether to include an edge between every pair of object locations in its trip (see variable $z_{i,j}$ in Alg. 4). However, in a practical mobile robotic visual sensing scenario, it is not very likely that the robot would need to solve the optimization problem with a very large number of sensing locations. For instance, some of the objects may be near each other and can thus share one sensing location. As such, the robot can group nearby objects and solve the planning problem with fewer locations, when the total number of objects in the scene is large.

5.7.3 Detection Networks

In this chapter, we did not use end-to-end detection architectures (e.g., Faster-RCNN [46]). This is because such detection networks tend to miss a lot of objects, especially in practical robotic settings where the visual recognition can be difficult due to non-ideal lighting, low resolution, small object size, and uncommon viewpoints, as we have observed in the early stage of this study. In fact, on the COCO detection leaderboard, the best method has an average recall of 0.727 for medium-sized objects, indicating that it can miss many objects. In addition, the commonly-used Faster-RCNN with ResNet-101 pipeline only has an average recall of 0.553.¹⁷ Using such detection models makes it difficult for the robot to improve its recognition performance, as it would not even discover several objects in the first place. Thus, in our robotic experiments, we utilize a saliency and depth-based method to discover potential objects near the robot, independent of the recognition difficulty for the onboard DCNN. This ROI extraction method works effectively for our campus experiments even though there are many visu-

¹⁷Recall is the fraction of the number of correctly identified/classified objects of a certain class over the total number of objects of that class, in an image. More details can be found at <http://cocodataset.org/#detection-eval>. The average recall of other detection models can be found at <http://cocodataset.org/#detection-leaderboard>.

ally hard-to-detect objects. Its performance, however, may degrade when there is a lot of visual clutter and/or the target objects are not salient, which may result in inaccurate localization or missing objects. Therefore, as part of future work, one could develop a localization/detection model that can discover visually-challenging targets across different scenarios, and integrate our CoMRF-based approach with it.

In this chapter, we showed how the robot can extract robust object similarity information from the output of its DCNN classifier, despite poor initial sensing, and exploit it for joint labeling, path planning and human query selection. In the next chapter, we focus on the communication cost in realistic channel environments and show how the robot can properly co-optimize its sensing, communication and motion planning under resource constraints.

Chapter 6

Co-Optimization of Motion, Communication, and Sensing via Monte Carlo Tree Search

In this chapter, we more explicitly consider the elements of path planning and communication, and adopt a more general robotic sensing model, in order to focus on the co-optimization of path planning, communication, and sensing in mobile robotic operations, in realistic wireless communication environments. Fig. 6.1 illustrates the general task scenario considered in this chapter. More specifically, the robot navigates from the start position to a given final position, and needs to sense a number of sites in the field. For each site, the robot must move within a certain distance in order to sense it. When visiting each site, the robot collects new sensing data. The robot is required to transmit all its collected data (and possibly some initial data) to the remote station by the end of the trip, while traversing the field. Its transmission energy cost is subject to a spatially-varying wireless communication channel that experiences fading. Our goal is to minimize the total motion and communication energy cost of this robotic operation, by co-optimizing the robot's sensing and path, as well as its data transmission as it traverses the path. This is a considerably challenging problem as it involves the combinatorial characteristics of path planning, the path constraints imposed by the sensing

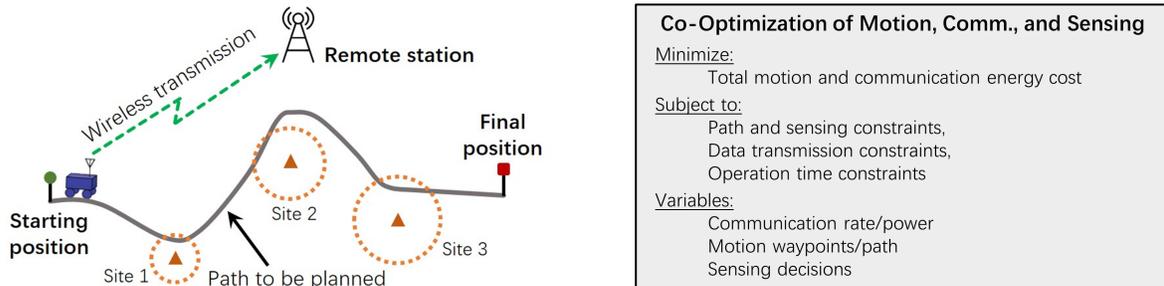


Figure 6.1: (Left) Example of the robotic task scenario considered in this chapter. (Right) High-level description of the co-optimization problem.

requirements, the spatially-varying wireless channel which cannot be modeled analytically, and the nonlinear transmission optimization. In order to solve this co-optimization problem, we propose a novel approach utilizing Markov Decision Process (MDP) and Monte Carlo Tree Search (MCTS), and show how it can solve this problem efficiently while providing theoretical convergence guarantees.

The rest of this chapter is organized as follows. Sec. 6.1 provides a primer on MDP and MCTS. In Sec. 6.2, we formulate the co-optimization problem and in Sec. 6.3, we discuss how to transform the co-optimization into an MDP and utilize MCTS to solve it. In Sec. 6.4, we prove the convergence and show the theoretical properties of our proposed approach. In Sec. 6.5, we validate our approach in a realistic 2D wireless channel environment.

6.1 Preliminaries

In this section, we provide overviews of Markov Decision Process [127] and Monte Carlo Tree Search [128].

6.1.1 Markov Decision Process

A Markov Decision Process (MDP) is a mathematical framework for modeling a discrete-time sequential decision-making process. Suppose that a decision-making agent is in some state. The goal of MDP is to find the optimum action to take in this state. An MDP is defined by four components.

State: There is a finite set of states S , where each $s \in S$ represents a state that the decision-making agent can be in. When the agent reaches a terminal state, it finishes the decision-making process.

Action: There is a finite set of actions Ω_A , where each $a \in \Omega_A$ is a feasible action that the agent can take. The action set can be state dependent, i.e., for each state s , the set of feasible actions $\Omega_{A,s}$ is different.

Transition: $\zeta(s, a, s')$ is the probability that by taking action a in state s , the agent moves to state s' . The MDP is deterministic if there is no randomness in the state transitions.

Reward: $w(s, a, s')$ is the immediate reward that the agent receives after transitioning from s to s' , due to action a . There can be a terminal reward $w_T(s_T)$ when the agent reaches a terminal state s_T .

In this chapter, we are interested in finite-horizon, deterministic, undiscounted MDPs, where the core problem is to find a decision policy $\pi : S \rightarrow \Omega_A$ that maximizes the cumulative reward of a state s , which is given by

$$X_s = \sum_{t=t_s}^{T-1} w(s_t, \pi(s_t), s_{t+1}) + w_T(s_T), \quad (6.1)$$

where t_s is the time instance when agent is in state s and T is the time step that the agent reaches a terminal state.

Given an initial state s_1 , the optimization for deriving the optimum decision policy

can be written as follows:

$$\max_{\pi(s)} \sum_{t=1}^{T-1} w(s_t, \pi(s_t), s_{t+1}) + w_T(s_T), \quad (6.2)$$

where we maximize the cumulative reward of s_1 , over the space of all possible decision policies.

There exist standard methods, such as value iteration and policy iteration, which can solve for the optimum decision for every state. However, they do not scale for large MDPs.

6.1.2 Monte Carlo Tree Search

Monte Carlo Tree Search (MCTS) is a popular algorithm that can be used to efficiently solve finite-horizon, large-scale MDPs. In this overview, we assume that the MDP is finite-horizon, deterministic, and undiscounted, which is the most relevant form for the problem-of-interest in this paper.

As discussed in Sec. 6.1.1, the goal of the MDP is to find a policy $\pi(s)$ that maximizes the cumulative reward. Denote the optimum cumulative reward of s as X_s^* . The optimum policy can then be recursively written as:

$$\pi^*(s) = \operatorname{argmax}_{a \in \Omega_{A,s}} w(s, a, s') + X_{s'}^*. \quad (6.3)$$

In theory, traditional methods such as value/policy iteration can be used to solve for the optimum policy. However, since they require iteratively computing the cumulative rewards for all the states, they quickly become computationally intractable for MDPs with a large state space. On the other hand, MCTS is an online algorithm that computes the optimum action for the current state and avoids computation over all the states.

More specifically, it constructs a tree to trace the future states that could be reached from the current state via one or several actions and biases the computation (i.e., Monte Carlo simulations) towards states that are more likely to produce large cumulative rewards. Meanwhile, it also methodically explores states with a small number of Monte Carlo samples. In this way, MCTS not only continuously refines the cumulative reward estimates of the promising states, but also reduces the chance of missing a good state due to inaccurate Monte Carlo estimation caused by insufficient sampling. By adopting a proper sampling rule that balances such exploitation and exploration, the expectation of the cumulative reward estimate of a state converges to the optimum efficiently and the probability of failing to choose the optimum action also converges to zero rapidly. These properties make it computationally favorable to apply MCTS to large MDPs for online decision-making.

Next, we briefly explain how MCTS works.¹ It starts with a tree containing only the root node τ_1 that represents the initial state s_1 . It then iteratively grows the tree, where each tree node uniquely represents one of the MDP states and each edge represents a state transition.² For each state s included in the tree, MCTS maintains a visit count, n_s , and an estimate of the optimum cumulative reward after n_s visits, \bar{X}_{s, n_s} .³ In each MCTS iteration, there are four main stages as described below.

Selection: In each iteration, MCTS performs a tree traversal from the root node τ_1 . During the traversal, at each node (i.e., a state of the MDP), MCTS selects an action according to some selection policy, which leads to the next node (more on this later). It continues this traversal until the selected action leads to a state s_{next} that does not have

¹When describing the MCTS process, we use “node” to refer to a tree node in the search tree and use “state” for a state of the corresponding MDP. We refer the readers to [128] for a comprehensive survey of MCTS.

²We assume that there is a one-to-one mapping from tree nodes to states. This is true in the problem studied in this chapter, since we use the robot’s path history as the state. It should, however, be noted that not all the states of the MDP are represented in the tree.

³ \bar{X}_{s, n_s} is set to 0 initially when there is no visit to state s yet (i.e., $n_s = 0$).

a corresponding node in the current tree. Denote the last traversed node in the current tree as τ_{last} .

Expansion: We add a new node τ_{next} representing s_{next} to the tree, as a child node of τ_{last} .

Simulation: Starting from s_{next} , random feasible actions are taken in the MDP until this process reaches a terminal state.

Back Up: Upon reaching a terminal state, we have obtained all the rewards in this particular decision process. As such, we can calculate the cumulative reward for any state s in the tree traversal of the current iteration: $X_{s, n_s+1} = \sum_{t=t_s}^{T-1} w(s_t, a_t, s_{t+1}) + w_T(s_T)$, where t_s is the time step when this tree traversal is in state s , T is the time when the process reaches the terminal state in this iteration, and n_s records the number of previous iterations where s was part of the traversal. Then, for each s in the current traversal, we update the estimate of the optimum cumulative reward as follows: $\bar{X}_{s, n_s+1} = (n_s \bar{X}_{s, n_s} + X_{s, n_s+1}) / (n_s + 1)$, where \bar{X}_{s, n_s} is the average value of the cumulative rewards of s seen in the past n_s iterations where s was part of the traversal and \bar{X}_{s, n_s+1} is the updated average cumulative reward. We also increment the visit count n_s by 1.

These four steps are repeated until the number of MCTS iterations reach a predefined computation budget, after which the decision policy for the initial state s_1 is: $a^\pi = \operatorname{argmax}_{a \in A_{s_1}} w(s_1, a, s') + \bar{X}_{s', n_{s'}}$. The agent takes an action according to this decision policy and transitions to the next state s_2 . In order to determine the action to take in s_2 , the agent then performs a new batch of MCTS iterations, with s_2 being the new initial state. This online planning process ends when the agent arrives in a terminal state.

In order to ensure efficient convergence, it is necessary to choose a proper selection policy while traversing down the tree in the first MCTS step. A common choice is based

on the Upper Confidence Bound (UCB) [129]:

$$a^\dagger = \operatorname{argmax}_{a \in A_s} w(s, a, s') + \bar{X}_{s', n_{s'}} + \sqrt{2 \ln(n_s) / n_{s'}}, \quad (6.4)$$

where action a^\dagger leads the traversal to the next tree level.⁴

It can be seen from Eq. 6.4 that the first two terms bias MCTS to select an action that is estimated to produce a large cumulative reward, while the third term allows MCTS to explore an action that leads to a less-visited state, reducing the chance of missing a state with a large cumulative reward due to insufficient sampling. As we shall see, using such a tree traversal strategy will result in an efficient convergence for our co-optimization problem.

6.2 Co-Optimization Problem Formulation

Consider the scenario where a robot has to travel from a start position, q_s , to a given destination, q_f , in a wireless channel environment, as shown in Fig. 6.1. During the trip, the robot needs to sense V sites in the field. In order to sense a site j , the robot must be within a sensing range $\tilde{\Delta}_j$ of it at some point during the trip, in order to collect Ξ_j bits of sensing data from this site. It also carries Ξ_0 bits of initial data when starting the operation. The robot is required to transmit all the collected data (and the initial data) to the remote station by the end of the trip, while minimizing its total motion and communication energy usage.

In the optimization formulation, we consider a discretized 2D workspace consisting of regular grids. The robot's path is defined by a sequence of waypoints, each being the

⁴Note that the third term in Eq. 6.4 can be infinity when $n_{s'} = 0$. When there is only one such s' , the policy chooses the action that leads to this s' . When there are multiple such next states, then it is common to simply choose one of the corresponding actions randomly. When $n_s = 1$ and $n_{s'} = 0$, the third term is also taken as infinity.

center of its corresponding grid. Two consecutive waypoints belong to two neighboring grid locations in the environment and determine a step in the path. Let $Q = [q_1, \dots, q_K]$ denote a path from the start position $q_1 = q_s$ to the final position $q_K = q_f$. Let $\tilde{q}_1, \dots, \tilde{q}_V$ denote the locations of the V sites to be sensed.⁵ The robot's path is subject to a total operation time budget: $\mathcal{T}(Q) \leq T$, where $\mathcal{T}(Q)$ is the total time needed to traverse the path Q and T is the time budget. Let $R = [r_1, \dots, r_K]$ denote the respective spectral efficiencies to be used at the corresponding waypoints $[q_1, \dots, q_K]$.⁶ The communication bandwidth B is constant during the operation and we denote $\tilde{\Xi}_j = \Xi_j/B, \forall j \in \{0, \dots, V\}$. The transmission time duration for each waypoint is t_c . The robot uses a constant speed u_{const} when moving. The motion energy cost of traversing the path Q is $\mathcal{E}_m(Q)$ and the communication energy cost is denoted by $\mathcal{E}_c(Q, R; \Upsilon)$, where Υ is the predicted channel over the space and $\Upsilon(q)$ is the predicted channel at location q , based on the channel prediction framework of Sec. 2.1. The co-optimization problem can then be formulated as follows:

$$\begin{aligned}
 & \min_{Q, R} \quad \mathcal{E}_m(Q) + \mathcal{E}_c(Q, R; \Upsilon) \\
 & \text{s.t.} \quad (1) \min_{i \in \{1, \dots, K\}} \|q_i - \tilde{q}_j\| \leq \tilde{\Delta}_j, \quad \forall j \in \{1, \dots, V\}, \\
 & \quad (2) \mathcal{T}(Q) \leq T, \quad (3) q_1 = q_s, \quad q_K = q_f, \\
 & \quad (4) \sum_{i=1}^K r_i t_c = \sum_{j=0}^V \tilde{\Xi}_j, \\
 & \quad (5) \sum_{i=1}^{k_j} r_i t_c \leq \sum_{z=0}^{j-1} \tilde{\Xi}_z, \quad \forall j \in \{1, \dots, V\}, \\
 & \quad (6) 0 \leq r_i \leq r_{\max}, \quad \forall i \in \{1, \dots, K\},
 \end{aligned} \tag{6.5}$$

⁵The order of visit for the sites is dictated by their order on the shortest path from the start position to the destination through these sites. This is the optimum order as long as no two sites are too close to each other.

⁶It should be noted that we do not consider quantization of the rates in our formulation. Our work can be easily extended to include this, by using the corresponding literature in the area of communications.

where the objective function is the total motion and communication energy cost, with the path and the spectral efficiencies to be optimized. Constraints (1)-(3) are related to the motion. Constraint (1) ensures that the robot moves within the sensing range for each site at some point in the path. Constraint (2) ensures that the total travel time does not exceed the time budget. Constraint (3) ensures that the path is from the start position to the destination. The remaining constraints are related to the transmission. Constraint (4) enforces the robot to transmit all the data by the end of the trip. Constraint (5) ensures a valid transmission plan, i.e., the data are transmitted only after they have been collected. In this constraint, $k_j = \min\{i \in \{1, \dots, K\} \mid \|q_i - \tilde{q}_j\| \leq \tilde{\Delta}_j\}$, $\forall j \in \{1, \dots, V\}$, which is the index of the first waypoint within the sensing range for site j . In other words, D_j are available after the robot has moved to q_{k_j} . Constraint (6) provides the feasible range for the spectral efficiencies. It should be noted that the robot minimizes the expectation of its communication energy cost in the objective since the predicted channel over the space is a multivariate random variable.

Due to the combinatorial nature of path planning and the coupling between motion and communication, problem (6.5) is a nonlinear, nonconvex, and combinatorial optimization problem, which is challenging to solve. Let's consider a simplified version first, where the robot's path, $Q = [q_1, \dots, q_K]$, is already given and we only need to optimize the spectral efficiencies. Based on the channel prediction model of Sec. 2.1, this sub-problem

can be written as follows:

$$\begin{aligned}
& \min_R \sum_{i=1}^K \frac{(2^{r_i} - 1)}{Z} \mathbb{E} \left[\frac{1}{\Upsilon(q_i)} \right] \cdot t_c, \\
& \text{s.t. (1)} \quad \sum_{i=1}^K r_i t_c = \sum_{j=0}^V \tilde{\Xi}_j, \\
& \quad \quad \quad \text{(2)} \quad \sum_{i=1}^{k_j} r_i t_c \leq \sum_{z=0}^{j-1} \tilde{\Xi}_z, \quad \forall j \in \{1, \dots, V\}, \\
& \quad \quad \quad \text{(3)} \quad 0 \leq r_i \leq r_{\max}, \quad \forall i \in \{1, \dots, K\},
\end{aligned} \tag{6.6}$$

where the objective function is the communication energy cost and the communication-related constraints are the same as in problem (6.5). Since the entire path Q is given, $\mathbb{E}[1/\Upsilon(q_i)]$ can be evaluated based on the predicted channel at location q_i . It can be seen that this is a convex optimization problem which can be solved very efficiently, as the objective function is convex and the constraints are linear [116].

This is an important observation that motivates our proposed approach to solve the co-optimization problem. If we can design an algorithm where the sensing/motion part and the transmission part are computed iteratively in different steps, then we can potentially make the problem tractable, as the decision space is considerably simpler when we only consider sensing/motion and the convex transmission sub-problem can be solved efficiently. In the next section, we show how we can indeed achieve this by properly formulating an MDP for the co-optimization problem (6.5) and utilizing MCTS to solve it.

6.3 Solving the Co-Optimization Problem via Monte Carlo Tree Search

The co-optimization problem (6.5) can be seen as a sequential decision-making problem, where at each step, the robot decides on a motion action and a transmission action such that its total energy cost is minimized. Markov Decision Process (MDP) thus provides a suitable framework for modeling this problem. In order to formulate a proper MDP (i.e., satisfying the Markov property), a state should not only include the robot's current waypoint, but also its past waypoints. This creates an exponentially large state space, making it infeasible to use traditional methods such as value/policy iteration. By utilizing Monte Carlo Tree Search (MCTS), on the other hand, we can efficiently handle the large state space. However, the coupled sensing-motion-transmission action space still presents a challenge since we cannot determine the optimum spectral efficiencies when the path is not fully determined, given the complex communication constraints of problem (6.5).

In order to resolve this issue, we propose a specially-designed MDP where we do not need to consider the transmission actions in the state transitions. More specifically, we only consider the motion actions in the state transitions, while the spectral efficiencies are optimized when evaluating the terminal reward in a terminal state. This allows us to iteratively optimize over the sensing/motion and the communication parts in two different stages of our proposed MCTS, and get around the challenge of determining the optimal spectral efficiencies without a complete path.

We next describe in details how we formulate this MDP.

6.3.1 Co-Optimization as a Markov Decision Process

The optimization problem (6.5) can be written as a deterministic, finite-horizon, and undiscounted MDP as follows:

State: Let $Q' = [q_1, \dots, q_k]$ denote a partial path, where $q_1 = q_s$ and q_k may not have reached q_f . The partial path contains the first part of a complete path from q_s to q_f . For instance, a partial path of $[q_1, q_2, q_3]$ indicates that the robot starts at $q_1 = q_s$ and will move to q_2 and q_3 sequentially, with the remaining path not yet determined. A partial path Q' then represents a state in this MDP. A state is a terminal state if the corresponding partial path has reached the destination, i.e., $q_k = q_f$.

Action: Given a state in this MDP with $Q' = [q_1, \dots, q_k]$, an action is represented by a next location that the robot can move to. The feasible set of actions is the set of those neighboring locations of q_k , after moving to which the robot can still reach the destination while satisfying the motion constraints on the entire path, i.e., constraints (1)-(3) of problem (6.5).

Transition: Given a feasible action, the next state is obtained by appending this location to the current partial path. For instance, given a current state of $[q_1]$ and an action of q_2 , the next state is given by $[q_1, q_2]$. There is no randomness in moving from one location to another.

Reward: We take the reward to be zero for any state transition. The terminal reward is then taken as: $w_T(Q) = -(\mathcal{E}_m(Q) + \min_{R \in \Omega_C} \mathcal{E}_c(Q, R; \Upsilon))$, where Q is the complete path from q_s to q_f associated with the terminal state and Ω_C is the feasible set of spectral efficiencies as defined by constraints (1)-(3) in problem (6.6). The first term (in parentheses) is the motion energy cost of traversing Q and the second term is the communication energy cost given by solving problem (6.6), which is taken as infinity if problem (6.6) is infeasible given Q . $-w_T(Q)$ is then the minimum total energy cost given

that the robot would use the full path Q .

It can be seen that we have moved the transmission optimization into the terminal reward computation and only the motion part is considered in the MDP actions. Next, we show that by solving this MDP, we also obtain the optimum solution to the original optimization problem (6.5).

Proposition 6.1 *By solving the MDP of Sec. 6.3.1 with the initial state $s_1 = [q_s]$, we obtain the optimum solution to the original co-optimization problem (6.5).*

Proof: As the MDP is undiscounted and there is no reward for any intermediate step, the cumulative reward of s_1 is given by $w_T(Q)$. By maximizing the cumulative reward of s_1 , we have the following optimization problem:

$$\underset{q_2, \dots, q_K}{\text{maximize}} w_T(Q) = -(\mathcal{E}_m(Q) + \min_{R \in \Omega_C} \mathcal{E}_c(Q, R; \Upsilon)), \quad (6.7)$$

where we optimize over the actions q_2, \dots, q_K , which also need to be feasible, i.e., the complete path $Q = [q_1, \dots, q_K]$ needs to satisfy constraints (1)-(3) of problem (6.5).

Since the MDP maximizes the negative value of the total energy cost, and the motion actions and spectral efficiencies already need to satisfy their respective constraints in the original problem (6.5), it can be easily seen that the optimum path Q^* and the optimum spectral efficiencies R^* , obtained from solving the MDP, are also optimal for problem (6.5). ■

6.3.2 Solution via Monte Carlo Tree Search

The MDP formulation of Sec. 6.3.1 facilitates applying MCTS. First, in order to solve problem (6.5), we do not need to derive the optimum decisions for all the MDP states, as many of them are sub-optimal and thus irrelevant to the original optimization problem. Instead, MCTS provides a methodical framework to bias the computation towards

promising states. Secondly, MCTS is well suited for the structure of the MDP. In each iteration, it only needs to perform the transmission optimization at the end of the simulation stage, when the simulation reaches a terminal state and the path becomes complete. This avoids the difficulty of determining the optimal spectral efficiencies without a complete path. Lastly, the convex optimization-based terminal reward evaluation (using problem (6.6)) allows for fast Monte Carlo simulations.

Next, we describe in details how to utilize MCTS to solve this MDP.

Selection Policy

As discussed in Sec. 6.1.2, we keep track of the average cumulative reward \bar{X}_{s, n_s} and a visit count n_s for each state s during the MCTS iterations. These quantities are used in the UCB-based selection policy for selecting the next state to move to during the tree traversal stage:

$$s^\dagger = \operatorname{argmax}_{s' \in C_s} \bar{X}_{s', n_{s'}} + \sqrt{2 \ln(n_s) / n_{s'}}, \quad (6.8)$$

where s^\dagger is the next state to move to and C_s is the set of next states reachable from s via a feasible action. Note that we select from the next states instead of the actions, since there is no intermediate reward and the transition is deterministic.

Expansion and Simulation

When adding a new node to the tree, the expansion part follows from the standard MCTS. In the simulation stage, a random feasible action (i.e., a random next waypoint) is taken until the path reaches the destination. More specifically, given a state with partial path $Q' = [q_1, \dots, q_k]$, the set of feasible next locations is the set of the neighboring locations of q_k , after moving to which the robot can visit the unsensed sites and reach the final position within the remaining time budget (i.e., satisfying constraints (1)-(3) of problem (6.5)). Denote the next site to sense as $\tilde{q}_{j'}$. We can obtain a random feasible next location by sampling from the set: $\Omega_{\text{next}} = \{q_{k+1} \mid \|\tilde{q}_{j'} - q_{k+1}\| \tau + \sum_{j=j'+1}^V \|\tilde{q}_j -$

$\tilde{q}_{j-1}\|_{\mathcal{T}} + \|q_f - \tilde{q}_V\|_{\mathcal{T}} \leq T - \mathcal{T}(Q') - \|q_{k+1} - q_k\|_{\mathcal{T}}$ and $q_{k+1} \in \Omega_{q_k}$ }, where $\|q_i - q_j\|_{\mathcal{T}}$ is the minimum time needed to move from q_i to q_j (on the grids) and Ω_{q_k} is the set of neighboring locations of q_k . It can be seen that any location from Ω_{next} allows the robot to visit the exact location of each unsensed site and then reach the destination within the remaining time, i.e., any location from this set is a feasible next location.⁷

Reward Evaluation and Back Up

The simulation ends when it encounters a terminal state, i.e., the partial path reaches the destination. In order to facilitate the convergence of MCTS, we need to transform the terminal reward $w_T(Q)$ into $[0, 1]$ (see Eq. 6.7 for the details of $w_T(Q)$) [129]. To do this, we utilize a simple baseline strategy where the robot moves along straight lines, sequentially from the start position to each exact site location and finally to the destination. We denote the negative total energy cost of this straight-path baseline as w_{sp} . We then take $\tilde{w}_T(Q) = \max(0, 1 - w_T(Q)/w_{sp})$ as the transformed reward, which is in $[0, 1]$. $\tilde{w}_T(Q)$ indicates the percentage total energy cost reduction over this baseline and is zero if the total cost resulted from Q is larger than that of the baseline.

The transformed reward $\tilde{w}_T(Q)$ is then backed up. For each state s that is part of the current tree traversal, its average cumulative reward is updated as follows:

$$\bar{X}_{s, n_s+1} = (\tilde{w}_T(Q) + n_s \bar{X}_{s, n_s}) / (n_s + 1) \quad (6.9)$$

and the visit count n_s is incremented by 1. In addition to this standard MCTS back-up procedure, for each s , we also record the maximum cumulative reward seen so far, as

⁷Note that this set does not contain all the possible feasible next locations, as there may exist a feasible next location that requires the robot to not visit the exact locations of the unsensed sites in order to reach the destination within the remaining time. However, finding such a feasible next location requires solving an optimization problem similar to the shortest-path traveling salesman problem with neighborhoods [130], which is computationally expensive for the simulation stage. On the other hand, sampling a point from Ω_{next} is quick, and as we shall see in Sec. 6.5, allows us to efficiently obtain near-optimal solutions to the complex co-optimization problem (6.5).

follows:

$$\widehat{X}_{s, n_s+1} = \max\{\widehat{X}_{s, n_s}, \tilde{w}_T(Q)\}, \quad (6.10)$$

where $\widehat{X}_{s, n_s} = 0$ initially when $n_s = 0$.

Solution Extraction

Given the initial state s_1 , we perform N_I MCTS iterations (N_I dictated by the computation budget). As there is no randomness in the rewards of this MDP, \widehat{X}_{s, n_s} records the best solution (i.e., the minimum total energy cost) seen so far given that the robot would traverse the partial path Q'_s . Therefore, instead of using \bar{X}_{s, n_s} as in standard MCTS, we use \widehat{X}_{s, n_s} to decide the next location that the robot should move to:

$$s^\pi = \operatorname{argmax}_{s' \in C_{s_1}} \widehat{X}_{s', n_{s'}}, \quad (6.11)$$

where the last waypoint in the partial path of s^π is the next location for the robot to move to and C_{s_1} is the set of next states reachable from s_1 via a feasible action.

We then set s^π as the new initial state and perform another N_I iterations, after which we select the best next state. This process ends when we reach a terminal state, where we obtain the best complete path, Q^* , and we can solve for the optimum spectral efficiencies, R^* . Q^* and R^* are then the solution to problem (6.5) given by our proposed approach.

6.4 Theoretical Analysis

In this section, we mathematically prove the convergence of our proposed approach and characterize its convergence speed. Furthermore, we mathematically characterize properties of the optimum solution.

6.4.1 Convergence

The first result shows the convergence of the expected average cumulative reward of a state.⁸ It should be noted that X_s^* is the optimum cumulative reward of state s in the following theorem, as defined in Sec. 6.1.2.

Theorem 6.1 *When MCTS with the UCB selection policy is applied to the MDP of Sec. 6.3.1, for any state s , the bias of \bar{X}_{s,n_s} is $O(\ln(n_s)/n_s)$, i.e., $\|\mathbb{E}[\bar{X}_{s,n_s}] - X_s^*\| = O(\ln(n_s)/n_s)$. Moreover, $X_s^* = \max_{s' \in C_s} X_{s'}^*$, where C_s is the set of next states reachable from state s via a feasible action, which corresponds to the set of child nodes of the node of state s in the tree.*

Proof: Note that the immediate reward is zero for any state transition. The results then follow directly from Theorem 2 and Theorem 6 of [129]. ■

Theorem 6.1 shows that the bias of the average cumulative reward of a state s is $O(\ln(n_s)/n_s)$. Moreover, the expected average cumulative reward converges to the optimum cumulative reward of the optimum next state. By the optimum next state of a state s , we mean the next state s^* which has the maximum optimum cumulative reward, i.e., $s^* = \operatorname{argmax}_{s' \in C_s} X_{s'}^*$.

In Theorem 6.1, X_s^* is characterized in a recursive manner. The next theorem then explicitly derives what the optimum cumulative reward is for each state.

Theorem 6.2 *When MCTS with the UCB selection policy is applied to the MDP of Sec. 6.3.1, for state s , $X_s^* = \max_{s' \in L_s} \tilde{w}_T(Q_{s'})$, where L_s is the set of terminal states with s being their common ancestor in the tree and $Q_{s'}$ is the complete path associated with the terminal state s' .*

⁸The expectation is taken over the randomness in the simulation stage of MCTS, as well as over the random selection when multiple next states have an infinitely-large second term in Eq. (6.8) due to $n_{s'} = 0$.

Proof: This can be shown by induction. First, the statement is true for a (sub)tree with only one node representing a terminal state. Then, consider a non-terminal state s in the tree. Assume that for each immediate next state $s' \in C_s$, $X_{s'}^* = \max_{s'' \in L_{s'}} \tilde{w}_T(Q_{s''})$. Based on Theorem 6.1, we then have $X_s^* = \max_{s' \in C_s} X_{s'}^* = \max_{s' \in C_s} \max_{s'' \in L_{s'}} \tilde{w}_T(Q_{s''}) = \max_{s'' \in L_s} \tilde{w}_T(Q_{s''})$, which completes the proof. ■

It can be seen from Theorem 6.2 that X_s^* is equal to the maximum terminal reward over all the terminal states that are reachable from state s via a sequence of feasible actions, which is indeed the optimum cumulative reward of s in this MDP.

As discussed in Sec. 6.3.2, unlike the standard MCTS, we use \hat{X}_{s, n_s} to decide the next waypoint for the robot to move to, as shown in Eq. 6.11. In the next results, we prove convergence properties for this decision policy. First, we show that the bias of the maximum cumulative reward converges to zero and is no greater than that of the average cumulative reward.

Theorem 6.3 *When MCTS with the UCB selection policy is applied to the MDP of Sec. 6.3.1, for any state s , $\|\mathbb{E}[\hat{X}_{s, n_s}] - X_s^*\| \leq \|\mathbb{E}[\bar{X}_{s, n_s}] - X_s^*\| = O(\ln(n_s)/n_s)$.*

Proof: It is always true that $\hat{X}_{s, n_s} \geq \bar{X}_{s, n_s}$. Thus, $\mathbb{E}[\hat{X}_{s, n_s}] \geq \mathbb{E}[\bar{X}_{s, n_s}]$. In addition, $\bar{X}_{s, n_s} \leq \hat{X}_{s, n_s} \leq X_s^*$. Therefore, we have $\|\mathbb{E}[\hat{X}_{s, n_s}] - X_s^*\| \leq \|\mathbb{E}[\bar{X}_{s, n_s}] - X_s^*\| = O(\ln(n_s)/n_s)$. ■

Next, we study the probability of failing to move to the optimum next state, by using the decision policy of Eq. 6.11. We derive an upper bound on this probability. This bound shows that this probability converges to zero and characterizes its convergence speed. We first present two results from [129], which will be used in our proof.

Theorem 6.4 *(Theorem 3 of [129]) There exists some positive constant ρ such that for state s and any of its immediate next states $s' \in C_s$, $n_{s'} \geq \rho \ln(n_s)$.*

Theorem 6.5 (Theorem 4 of [129]) For state s , the following bounds hold for any given $\delta > 0$, provided that n_s is sufficiently large: $\mathbb{P}(\bar{X}_{s, n_s} \leq \mathbb{E}[\bar{X}_{s, n_s}] - 9\sqrt{2n_s \ln(2/\delta)}/n_s) \leq \delta$ and $\mathbb{P}(\bar{X}_{s, n_s} \geq \mathbb{E}[\bar{X}_{s, n_s}] + 9\sqrt{2n_s \ln(2/\delta)}/n_s) \leq \delta$.

Theorem 6.4 shows that the number of visits for a state is lower-bounded by a function of the number of visits of its parent node. This is because whenever a state is less visited, the selection policy of Eq. 6.8 will encourage the tree traversal to visit this state. Theorem 6.5 shows how the average cumulative reward of a state concentrates to its expectation probabilistically, as the number of visit increases.

Now we are ready to prove the convergence of the failure probability and characterize its convergence speed.

Theorem 6.6 For any state s , by using the maximum cumulative reward-based decision policy of Eq. 6.11, the probability of failing to reach the optimum next state, $\mathbb{P}(s^\pi \neq s^*)$, satisfies the following inequality, provided that n_s is sufficiently large:

$$\mathbb{P}(s^\pi \neq s^*) \leq \tilde{\rho}/\ln(n_s), \quad (6.12)$$

where s^* is the optimum next state (i.e., $s^* = \operatorname{argmax}_{s' \in C_s} X_{s'}^*$), s^π is the next state given by the decision policy of Eq. 6.11, and $\tilde{\rho}$ is a positive constant.

Proof: First, it can be easily confirmed that the following inequality holds:

$$\mathbb{P}(s^\pi \neq s^*) \leq \sum_{s' \in C_s \setminus \{s^*\}} \mathbb{P}(\hat{X}_{s', n_{s'}} \geq \hat{X}_{s^*, n_{s^*}}). \quad (6.13)$$

Assume that $X_{s^*}^* > X_{s'}^*, \forall s' \in C_s \setminus \{s^*\}$, i.e., there exists only one optimum next state. Since $X_{s^*}^* - \mathbb{E}[\bar{X}_{s^*, n_{s^*}}] = O(\ln(n_{s^*})/n_{s^*})$, as shown in Theorem 6.1, there exists a sufficiently large N_1 such that when $n_{s^*} \geq N_1$, $\mathbb{E}[\bar{X}_{s^*, n_{s^*}}] - X_{s^*}^* \geq h_{s'}$, for all $s' \in C_s \setminus \{s^*\}$, where $h_{s'} = (X_{s^*}^* - X_{s'}^*)/2$.

Given $n_{s^*} \geq N_1$, if $\widehat{X}_{s', n_{s'}} \leq X_{s'}^*$ and $\widehat{X}_{s^*, n_{s^*}} > \mathbb{E}[\bar{X}_{s^*, n_{s^*}}] - h_{s'}$, then $\widehat{X}_{s', n_{s'}} < \widehat{X}_{s^*, n_{s^*}}$, which leads to

$$\mathbb{P}(\widehat{X}_{s', n_{s'}} \geq \widehat{X}_{s^*, n_{s^*}}) \leq \mathbb{P}(\widehat{X}_{s', n_{s'}} > X_{s'}^*) + \mathbb{P}(\widehat{X}_{s^*, n_{s^*}} \leq \mathbb{E}[\bar{X}_{s^*, n_{s^*}}] - h_{s'}),$$

where it can be easily seen that $\mathbb{P}(\widehat{X}_{s', n_{s'}} > X_{s'}^*) = 0$. Furthermore, since $\bar{X}_{s^*, n_{s^*}} \leq \widehat{X}_{s^*, n_{s^*}}$, we have

$$\mathbb{P}(\widehat{X}_{s^*, n_{s^*}} \leq \mathbb{E}[\bar{X}_{s^*, n_{s^*}}] - h_{s'}) \leq \mathbb{P}(\bar{X}_{s^*, n_{s^*}} \leq \mathbb{E}[\bar{X}_{s^*, n_{s^*}}] - h_{s'}).$$

By setting $\delta = 1/n_{s^*}$ for state s^* by using the result of Theorem 6.5, we have

$$\mathbb{P}(\bar{X}_{s^*, n_{s^*}} \leq \mathbb{E}[\bar{X}_{s^*, n_{s^*}}] - 9\sqrt{2\ln(2n_{s^*})/n_{s^*}}) \leq 1/n_{s^*},$$

where $9\sqrt{2\ln(2n_{s^*})/n_{s^*}}$ converges to zero as n_{s^*} goes to infinity. As such, there exists a sufficiently large N_2 such that when $n_{s^*} \geq N_2$, we have $9\sqrt{2\ln(2n_{s^*})/n_{s^*}} \leq h_{s'}, \forall s' \in C_s \setminus \{s^*\}$. Therefore, given $n_{s^*} \geq N_2$, we have the following for each $s' \in C_s \setminus \{s^*\}$:

$$\begin{aligned} & \mathbb{P}(\bar{X}_{s^*, n_{s^*}} \leq \mathbb{E}[\bar{X}_{s^*, n_{s^*}}] - h_{s'}) \\ & \leq \mathbb{P}(\bar{X}_{s^*, n_{s^*}} \leq \mathbb{E}[\bar{X}_{s^*, n_{s^*}}] - 9\sqrt{2\ln(2n_{s^*})/n_{s^*}}) \leq \frac{1}{n_{s^*}} \leq \frac{1}{\rho \ln(n_s)}, \end{aligned}$$

where the last inequality is based on Theorem 6.4.

Based on Eq. 6.13, we then have $\mathbb{P}(s^\pi \neq s^*) \leq \tilde{\rho}/\ln(n_s)$, where $\tilde{\rho}$ is a constant that depends on ρ and the size of C_s . Moreover, in order to have $n_{s^*} \geq \max\{N_1, N_2\}$, we need $n_s \geq \max\{e^{N_1/\rho}, e^{N_2/\rho}\}$, based on Theorem 6.4. Therefore, given n_s sufficiently large, the inequality of Eq. 6.12 holds. ■

As we can see in the above theorem, for a state s , the failure probability is upper-

bounded by a function of n_s , which is $O(1/\ln(n_s))$. In other words, the failure probability converges to zero at least as fast as $\tilde{\rho}/\ln(n_s)$, as n_s increases.

6.4.2 Properties of the Optimum Solution

In this part, we characterize properties of the optimum spectral efficiencies. First, we define two waypoints q_{i_1} and q_{i_2} to be in the same *segment* of the path if they satisfy one of the following three conditions:

$$\begin{aligned} (1) \quad & k_{j-1} < i_1, i_2 \leq k_j, \text{ for some } j \in \{2, \dots, V\}, \\ (2) \quad & i_1, i_2 \leq k_1, \quad (3) \quad i_1, i_2 > k_V, \end{aligned} \tag{6.14}$$

where $k_j = \min\{i \in \{1, \dots, K\} \mid \|q_i - \tilde{q}_j\| \leq \tilde{\Delta}_j\}$, $\forall j \in \{1, \dots, V\}$. In other words, if two waypoints belong to the same segment of the path, then the robot does not collect new data when traveling between these two points.

We now characterize the optimum transmission spectral efficiencies for problem (6.5). Note that in the following theorem, $\Upsilon(q)$ is the random variable describing the predicted channel at location q , as defined in Sec. 2.1.

Theorem 6.7 *Consider problem (6.5). Given a complete path Q , the optimum transmission spectral efficiencies satisfy the following properties:*

(1) $r_i^* \geq r_j^*$, if $\mathbb{E}[1/\Upsilon(q_i)] \leq \mathbb{E}[1/\Upsilon(q_j)]$ and q_i, q_j belong to the same segment of the path;

(2) $r_i^* = 0$, if $\mathbb{E}[1/\Upsilon(q_i)]$ is above a certain threshold;

(3) $r_i^* = r_{max}$, if $\mathbb{E}[1/\Upsilon(q_i)]$ is below a certain threshold.

Proof: Given a complete path Q , problem (6.5) then reduces to problem (6.6). Assume $\Xi_0 > 0$ and the robot does not need to use the maximum spectral efficiency all the time. There exists some strictly-feasible solution to problem (6.6), indicating that

Slater's condition holds [116]. As problem (6.6) is convex and Slater's condition holds, the Karush-Kuhn-Tucker (KKT) conditions are sufficient and necessary for optimality [116]. The Lagrangian is then as follows, where $\xi \succeq 0$, $\nu \succeq 0$, $\eta \succeq 0$, and λ are the dual variables:

$$\begin{aligned} \mathcal{L}(R, \xi, \nu, \eta, \lambda) = & \sum_{i=1}^K \frac{(2^{r_i} - 1)}{Z} \mathbb{E} \left[\frac{1}{\Upsilon(q_i)} \right] + \xi_i (r_i - r_{\max}) - \nu_i r_i \\ & + \sum_{j=1}^V \eta_j \left(\sum_{i=1}^{k_j} r_i - \frac{1}{t_c} \sum_{z=0}^{j-1} \tilde{\Xi}_z \right) - \lambda \left(\sum_{i=0}^K r_i - \frac{1}{t_c} \sum_{j=0}^V \tilde{\Xi}_j \right). \end{aligned} \quad (6.15)$$

For the KKT conditions, in addition to primal and dual feasibility, we have the gradient condition and complementary slackness as follows, with the optimum variables marked by \star . $\forall i \in \{1, \dots, K\}$, we have

$$\begin{aligned} \nabla_{r_i^*} \mathcal{L} = & \frac{2^{r_i^*} \ln(2)}{Z} \mathbb{E} \left[\frac{1}{\Upsilon(q_i)} \right] + \xi_i^* - \nu_i^* + \sum_{j \in \mathcal{V}_i} \eta_j^* - \lambda^* = 0, \\ \xi_i^* (r_i^* - r_{\max}) = & 0, \quad \nu_i^* r_i^* = 0, \end{aligned} \quad (6.16)$$

where $\mathcal{V}_i = \{j \mid j \in \{1, \dots, V\} \text{ and } k_j \geq i\}$. We also have the following additional complementary slackness conditions due to constraint (2) of problem (6.6):

$$\eta_j^* \left(\sum_{i=1}^{k_j} r_i^* - \frac{1}{t_c} \sum_{z=0}^{j-1} \tilde{\Xi}_z \right) = 0, \quad \forall j \in \{1, \dots, V\}. \quad (6.17)$$

Denote $\tilde{\eta}_i = \sum_{j \in \mathcal{V}_i} \eta_j^*$ from the KKT conditions. Then, the optimum spectral efficiencies can be derived as follows:

$$r_i^* = \begin{cases} 0, & \text{if } \mathbb{E}[1/\Upsilon(q_i)] \geq \frac{(\lambda^* - \tilde{\eta}_i)Z}{\ln(2)}, \\ r_{\max}, & \text{if } \mathbb{E}[1/\Upsilon(q_i)] \leq \frac{(\lambda^* - \tilde{\eta}_i)Z}{2^{r_{\max}} \ln(2)}, \\ \log_2 \left(\frac{(\lambda^* - \tilde{\eta}_i)Z}{\ln(2) \mathbb{E}[1/\Upsilon(q_i)]} \right), & \text{otherwise.} \end{cases} \quad (6.18)$$

As $\tilde{\eta}_i$ is the same for any two waypoints in the same segment of the path, it can be confirmed that the three properties stated in this theorem hold based on Eq. 6.18. ■

Theorem 6.7 shows that within the same segment of the path, the optimum spectral efficiency should be higher (lower) where the channel quality is better (worse). The channel quality is measured by $\mathbb{E}[1/\Upsilon(q_i)]$, which is lower (higher) for a better (worse) channel. Moreover, when the channel quality is better than a certain threshold, the robot should take the maximum spectral efficiency and when the channel quality is worse than a certain threshold, there should be no transmission.

Consider a data intensive case, where the remaining onboard data to be transmitted is non-zero all throughout the path, up to the last segment, with no restriction on the last segment. Then, property (1) of Theorem 6.7 holds for any two waypoints in the path and the optimum spectral efficiencies can be solved using bisection, as we show next.

Corollary 6.1 *When the remaining data is consistently non-zero before the robot senses the last site, the optimum spectral efficiencies also satisfy: $r_i^* \geq r_j^*$, if $\mathbb{E}[1/\Upsilon(q_i)] \leq \mathbb{E}[1/\Upsilon(q_j)]$, for any two waypoints in the path. Moreover, the optimum spectral efficiencies can be solved using bisection.*

Proof: When the remaining data is always non-zero before the robot visits the last site, we have $\eta_j^* = 0$, $\forall j \in \{1, \dots, V\}$ based on the complementary slackness conditions of Eq. 6.17. The optimum spectral efficiencies are then as follows:

$$r_i^* = \begin{cases} 0, & \text{if } \mathbb{E}[1/\Upsilon(q_i)] \geq \frac{\lambda^* Z}{\ln(2)}, \\ r_{\max}, & \text{if } \mathbb{E}[1/\Upsilon(q_i)] \leq \frac{\lambda^* Z}{2^{r_{\max}} \ln(2)}, \\ \log_2 \left(\frac{\lambda^* Z}{\ln(2) \mathbb{E}[1/\Upsilon(q_i)]} \right), & \text{otherwise.} \end{cases} \quad (6.19)$$

It can then be easily confirmed that the property stated in this corollary is true. Next, we show how to obtain the optimum spectral efficiencies via bisection. Based on Eq. 6.19, we have $r_i^* = \max(0, \min(r_{\max}, \log_2 \left(\frac{\lambda^* Z}{\ln(2) \mathbb{E}[1/\Upsilon(q_i)]} \right)))$, $\forall i \in \{1, \dots, K\}$. Due to constraint (4) of problem (6.5), $\sum_{i=0}^K r_i^* - \frac{1}{t_c} \sum_{j=0}^V \tilde{D}_j = 0$ needs to hold. Therefore, λ^* is the solution to the following equation:

$$\sum_{i=0}^K \max(0, \min(r_{\max}, \log_2 \left(\frac{\lambda^* Z}{\ln(2) \mathbb{E}[1/\Upsilon(q_i)]} \right))) - \frac{1}{t_c} \sum_{j=0}^V \tilde{\Xi}_j = 0,$$

where λ^* can be solved via bisection. Given λ^* , we can then calculate the optimum spectral efficiencies from Eq. 6.19. ■

Next, we consider a special case where the robot does not need to sense any sites. This captures several real-world robotic motion and communication scenarios. The corresponding co-optimization problem can be formulated as follows:

$$\begin{aligned} & \min_{Q, R} \mathcal{E}_m(Q) + \mathcal{E}_c(Q, R; \Upsilon) \\ & \text{s.t. (1) } \mathcal{T}(Q) \leq T, \quad (2) \quad q_1 = q_s, \quad q_K = q_f, \\ & \quad (3) \quad \sum_{i=1}^K r_i t_c = \tilde{\Xi}_0, \quad (4) \quad 0 \leq r_i \leq r_{\max}, \quad \forall i \in \{1, \dots, K\}, \end{aligned} \quad (6.20)$$

where the robot only needs to navigate from the start position to the destination and transmit the Ξ_0 initial data.

The next corollary characterizes the optimum transmission spectral efficiencies in this

special case.

Corollary 6.2 *Consider the special case shown in problem (6.20). Given a complete path Q , the optimum transmission spectral efficiencies satisfy all the properties in Theorem 6.7 and Corollary 6.1, and can be solved using bisection.*

Proof: This can be proved by analyzing the KKT conditions, similar to the proof of Corollary 6.1. ■

6.5 Simulation Experiments

In this section, we solve the co-optimization problem (6.5) in a realistic 2D wireless channel environment using our proposed approach. We first consider a scenario that involves sensing, communication, and motion, and present the solution obtained by using our proposed co-optimization approach. We further extensively compare it with a benchmark method that separately optimizes motion and communication. More specifically, the benchmark first computes the shortest path that satisfies the motion constraints of problem (6.5) and subsequently, given the path, optimizes the transmission along this path.⁹ Finally, we consider the special case of problem (6.20) with no sensing and further compare our approach with the best related state-of-the-art method.

6.5.1 Co-Optimizing Sensing, Communication, and Motion

We validate our proposed approach using a realistic simulated 2D wireless channel environment [118], where the channel parameters (obtained from real wireless measurements [1]) are: $\hat{\theta}_{ch,dB} = [-41.34, 3.86]$, $\hat{\alpha}_{ch,dB} = 3.20$, $\hat{\beta}_{ch} = 3.09$ m, and $\hat{\sigma}_{ch,dB} = 1.64$.

⁹Obtaining the shortest path that satisfies the motion constraints in problem (6.5) requires solving a traveling salesman problem with neighborhoods, and with the start and final positions directly connected in the tour. Given such a tour, we then remove the edge between the start and final positions in order to obtain the path from the start position to the final one. We utilize a self-organizing map-based algorithm to compute such a path [130].

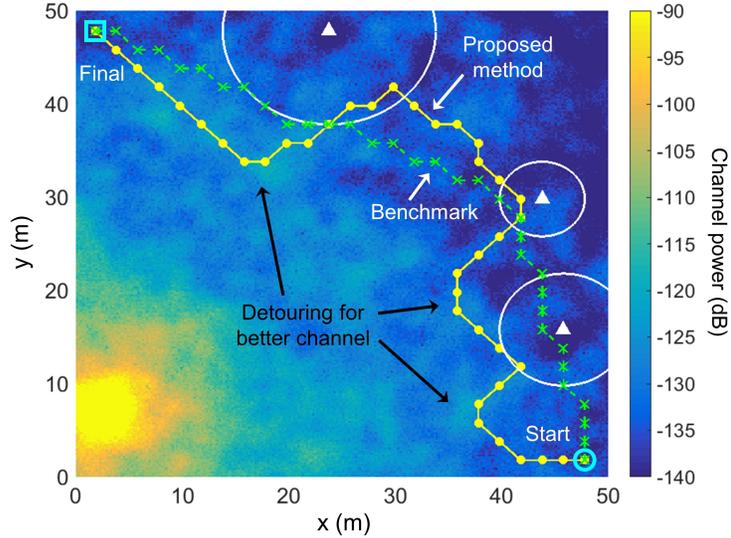


Figure 6.2: Paths from solving problem (6.5) with three sites to sense, by using our proposed approach (yellow) and the benchmark (green). The yellow dots and solid curve represent the waypoints and the path, respectively, as given by our proposed approach. The green crosses and dashed curve represent the waypoints and the path given by the benchmark method. The circle and square indicate the start and final positions, respectively. The white triangles indicate the site locations and the white circles indicate the sensing ranges for the sites. The colormap indicates the true channel power over this environment, where brighter (darker) colors indicate higher (lower) channel qualities. See the color pdf for optimal viewing.

The robot predicts the channel with 1% prior channel measurements from random locations in this environment, based on the prediction framework of Sec. 2.1. The required BER is 10^{-6} . The communication bandwidth is 20 MHz. The receiver noise power is -100 dBm. The maximum spectral efficiency is 6 bits/s/Hz. The motion parameters are: $\kappa_1 = 7.4$ and $\kappa_2 = 0.29$, based on real power measurements of a robot [2]. The robot uses a constant speed of 1 m/s. The workspace is 50 m \times 50 m with a grid size of 2 m \times 2 m. We use the eight-neighbor setting: the robot can move to one of the eight neighboring grids that is within the workspace from its current grid in one step.

In this experiment, the robot’s starting position is [48, 2] and the final position is [2, 48]. The three sites are located at [46, 16], [44, 30], and [24, 48], respectively, and the sensing ranges for them are 6 m, 4 m, and 10 m, respectively. The robot has an initial

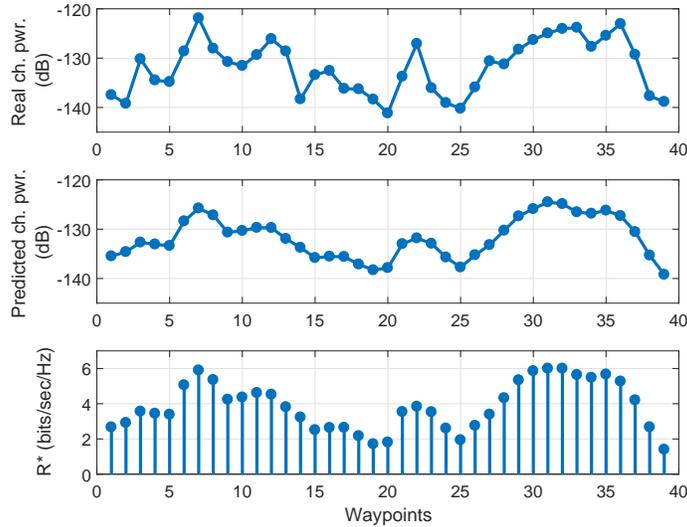


Figure 6.3: The optimal transmission spectral efficiencies from solving problem (6.5) with three sites to sense, by using our proposed approach. The first and second rows show the actual and the predicted channel powers at the waypoints, respectively. The third row shows the optimum spectral efficiencies for the waypoints.

Methods	\mathcal{E}_c	\mathcal{E}_m	$\mathcal{E}_c + \mathcal{E}_m$
Benchmark method	27,008 J	576 J	27,585 J
Proposed approach	9,142 J	818 J	9,960 J
Overall energy saving			55%

Table 6.1: Average energy costs by using our proposed approach and the benchmark method, over 50 random problem instances. The second, third, and fourth columns show the communication, motion, and total energy costs (in Joules), respectively. The last row shows the total energy saving by using our proposed approach, as compared to the benchmark.

data load of 60 bits/Hz and needs to transmit 30 bits/Hz additional data after sensing each site. The robot has a total time budget of 110 s. Fig. 6.2 shows the resulted path (yellow solid curve) by using our proposed approach. It can be seen that the robot detours towards areas with better channel qualities as needed (indicated by the black arrows in Fig. 6.2). Even on the path from site 2 to site 3, where the channel quality is generally poor, the robot is still able to detour a bit to exploit a slightly better channel (see the color pdf). Fig. 6.3 then shows the spectral efficiencies along the path using our proposed

approach. It can be seen that the robot adopts a higher (lower) spectral efficiency when the channel quality is better (worse). As the remaining data to be transmitted is never zero before the last site, this confirms the theoretical result of Corollary 6.1. On the other hand, since the benchmark separately optimizes the path and the transmission, its path (green dashed curve in Fig. 6.2) is unaware of the channel. This makes the robot traverse in areas with poor channel qualities, resulting in a large total energy cost of 10,910 Joules. In contrast, the total energy cost by using our proposed approach is 4,757 Joules, which is 56% lower. It takes our approach 47.59 s to solve the co-optimization in this case (on a 3.40 GHz i7 PC).

Next, we more extensively compare the performance of our proposed approach with the benchmark, over 50 problem instances. In these problem instances, the number of sites ranges from 1 to 4. In each problem instance, we use a different realization of the channel, and the site locations, sensing ranges, and the amount of data to be transmitted are randomized. Table 6.1 shows the energy costs (in Joules) by using our proposed approach and the benchmark, respectively, averaged over the 50 problem instances. It can be seen that overall, our proposed approach significantly reduces the total energy cost by 55%, as compared to the benchmark. This is because our approach is able to properly co-optimize sensing/motion and communication. For instance, our approach significantly reduces the communication cost by having the robot detour to areas with a better channel quality as needed, at the expense of a slightly higher motion cost.

6.5.2 Special Case of Co-Optimization without Sensing

We now consider the special case of problem (6.20), where there is no sensing. The 2D environment, the motion and communication parameters, and the number of MCTS iterations for each step are the same as in Sec. 6.5.1. In this experiment, the robot starts

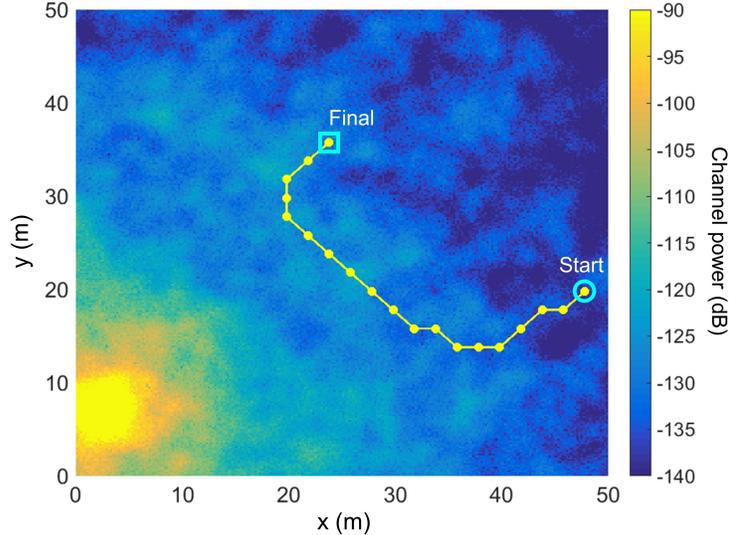


Figure 6.4: The resulting path when using our proposed approach to solve problem (6.5) for the case of no sensing. The yellow curve shows the path and the yellow dots indicate the waypoints. The circle and square indicate the starting and final positions, respectively. The colormap indicates the true channel power over this environment, where brighter (darker) colors indicate higher (lower) channel qualities. See the color pdf for optimal viewing.

from the initial position $[24, 36]$ and plans a path to the destination $[48, 20]$, with an operation time budget of 55 s. The robot needs to transmit a total of 80 bits/Hz initial data to the remote station by the end of the trip.

Fig. 6.4 shows the resulting path by using our proposed approach. It can be seen that the robot detours into a region with a better channel quality, before finally reaching the destination. Fig. 6.5 shows the optimum spectral efficiencies along this path given by our proposed approach, which are higher (lower) for waypoints with a better (worse) channel quality. This confirms the theoretical results of Corollary 6.2. In this experiment, the total energy cost by using our proposed approach is 1,642 Joules, while the benchmark costs 10,721 Joules, which is over $6\times$ as much as that of our proposed approach. It takes our approach 16.73 s to compute the solution in this case.

While for the general case with sensing, there is no existing approach that can co-optimize motion, communication, and sensing, for the case of no sensing, the work of [44]

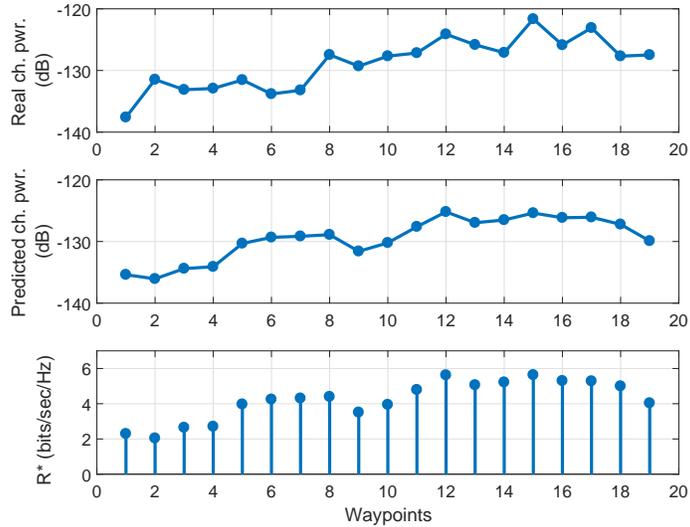


Figure 6.5: The optimum transmission spectral efficiencies from solving problem (6.5) when there is no sensing, by using our proposed approach. The first and second rows show the actual and the predicted channel powers at the waypoints, respectively. The third row shows the optimum spectral efficiencies for the waypoints.

co-optimizes motion and communication using a different approach. As compared to our proposed approach, [44] does not guarantee convergence to the optimum solution, has a higher total energy cost, and requires more computation time. For instance, in this experiment, the total energy cost by using [44] is 3,140 Joules, which is almost $2\times$ as much as that of our approach. Their algorithm takes 43.92 s to compute the solution, which is over $2.5\times$ slower than ours.

Overall, these results demonstrate that our proposed approach is capable of co-optimizing sensing, communication, and motion for the complex optimization problem (6.5). Furthermore, our approach considerably outperforms the benchmark, which separately optimizes motion and communication.

Chapter 7

Conclusions and Future Work

In this dissertation, we studied how to properly incorporate human collaboration and the robot’s visual understanding into the optimization of mobile robotic operations, in order to improve the robot’s visual recognition performance in challenging task scenarios. We first explicitly considered the aspect of human collaboration in mobile robotic operations. In order for the robot to properly ask the remote human operator for help with visual tasks, we developed a Deep Convolutional Neural Network (DCNN)-based approach to accurately predict human visual performance and showed how to incorporate such human prediction into the optimization of the robot’s field decision-making, in terms of sensing, motion, and querying/communication.

Next, we showed how the robot can infer and exploit object similarity, despite poor sensing, to improve its scene understanding. More specifically, while the initial classification performance may be poor for the robot’s DCNN classifier, we showed that the DCNN’s output still contains useful and reliable information about the similarity among the objects, which we showed how to capture. We then constructed a Markov Random Field (MRF), based on our inferred object similarities, and incorporated it into the optimization of the robot’s sensing, motion, and querying, which reduced the robot’s sensing burden and allowed it to significantly improve its visual performance.

Finally, we more explicitly took into account motion and communication in order to

focus more on the co-optimization of the robot’s motion, communication, and sensing in realistic wireless channel environments. We formulated a comprehensive optimization of the robot’s field operation that involved planning its full path, its sensing, as well as its data transmission along the path. In order to solve this complex optimization, we proposed a specially-formulated Markov Decision Process (MDP) and utilized Monte Carlo Tree Search (MCTS) to optimally and efficiently solve it.

We next summarize the results of each chapter and suggest possible future extensions.

7.1 Prediction of Human Visual Performance

In Chapter 3, we proposed a machine-learning based approach that allows the robot to probabilistically predict human visual performance for any sensory input. Equipped with this tool, we then showed how the robot can optimize its field decisions, in terms of asking for help, further sensing, and relying on itself. We tested the proposed approach on our campus with a number of robotic surveillance experiments and showed a considerable improvement in the performance. Moreover, we ran a larger-scale evaluation, with real data/human feedback, in a simulation environment to further showcase the effectiveness of the approach.

While we focused on robotic surveillance tasks based on imagery inputs, the methodology can be extended to other collaborative tasks/sensory inputs. The predictor can also be finetuned to the performance of a particular human operator for a longer-term partnership, or to the time of the day, among other factors, as part of future work.

7.2 Optimization of Human-Robot Collaboration with Human Performance Prediction

In Chapter 4, given the predicted human performance, we studied a general scenario of human-robot collaborative site inspection and target classification. More specifically, we considered the realistic case where the human visual performance is not perfect, depending on the sensing quality, and the robot has constraints in communication with the human operator. Furthermore, the robot has a limited onboard motion and communication energy budget and has to operate in realistic channel environments that experience path loss, shadowing, and multipath fading. We then showed how to co-optimize motion, sensing, and queries in human-robot collaborative site inspections, under limited motion and communication resources, constraints in communication with the operator, and imperfect human visual performance. We considered two types of realistic communication constraints: 1) the robot is given a limited number of chances to query the human operator, and 2) the wireless channel quality is not good enough all over the workspace to result in a reliable communication link, necessitating the robot to optimally choose the proper locations for communication with the operator.

Given a probabilistic human performance characterization and a probabilistic prediction of channel quality in realistic environments, we formulated the resulting co-optimization as Multi-Dimensional Multiple-Choice Knapsack Problems (MMKPs). We then proposed a Linear Program (LP)-based efficient near-optimal solution to the NP-hard MMKP, and mathematically characterized the optimality gap, showing that it can be considerably small. We also mathematically characterized several properties of the optimal solution. Finally, we validated the proposed approach comprehensively with extensive real human data, as well as with real wireless channel data from downtown San Francisco. The numerical results show that our proposed approach properly co-

optimizes motion, sensing, and human queries, and can significantly outperform benchmark methodologies in terms of task performance and resource savings.

7.3 Exploiting Object Similarity for Robotic Visual Recognition

In Chapter 5, we showed how the output of a DCNN classifier carries robust information on object similarity, even when each object is individually misclassified. More specifically, we introduced a training-free object similarity measure, based on the correlation of the feature vectors of the output of a DCNN classifier, to improve robotic visual classification under limited resources. We first probabilistically analyzed the correlation coefficient of the feature vectors of a pair of images from the output of an already-trained DCNN classifier, showing that it provides robust information on their similarity, without requiring any additional training (This validation was based on using a large dataset). Based on this analysis, we built a correlation-based Markov Random Field (CoMRF) for joint object labeling. Given a query budget and a motion budget, we then proposed a query-motion co-optimization framework to jointly optimize the robot’s query, path, and visual labeling, based on our CoMRF. By using a large COCO-based test set, a large-scale drone imagery dataset, and a large indoor scene dataset, our extensive evaluations showed that our proposed object similarity metric and the resulting CoMRF-based joint labeling and co-optimization methodology significantly improves the overall classification performance. Our several real-world robotic experiments on our campus further showcased the superior performance of our proposed CoMRF-based query-motion co-optimization approach.

As part of future work, it is possible to combine the proposed object similarity metric

with other types of visual scene understanding and correlation information (e.g., geometric relationship, object co-occurrence), in order to further enhance the robot’s visual recognition performance.

7.4 Co-Optimization of Motion, Communication, and Sensing via Monte Carlo Tree Search

In Chapter 6, we adopted a higher-level sensing model for the robot and studied a comprehensive co-optimization of a robot’s sensing, communication, and motion in a realistic wireless channel environment. More specifically, we considered a comprehensive scenario where the robot navigates from a start position to a destination. During the trip, it needs to sense some sites, by moving close enough near each site. The robot collects data when sensing each site and needs to transmit all collected data to a remote station or human operator by the end of its trip, under time/energy constraints. Our goal is to minimize the total motion and communication energy cost by co-optimizing the robot’s path, its data transmission along the path, and its sensing decisions.

In order to solve this complex optimization problem, we proposed a novel approach where we transformed the co-optimization problem into a specially-designed Markov Decision Process and utilized Monte Carlo Tree Search (MCTS) to solve it. More specifically, we showed that by iteratively optimizing the sensing/motion and the communication parts in different stages of MCTS, we can equivalently solve the original challenging co-optimization problem very efficiently. We then mathematically proved the convergence of our proposed approach, and characterized its convergence speed as well as other key properties of the optimum solution. Finally, we demonstrated the efficacy of our proposed approach in a realistic 2D wireless channel environment via extensive simulations.

Appendices

Appendix A

A.1 Multiple-Choice Multidimensional Knapsack Problems

In this section, we briefly introduce the Multiple-Choice Multidimensional Knapsack Problem (MMKP) [45], which is a complex variant of the standard Knapsack Problem (KP).¹ In the subsequent sections, we then show how our resource-constrained human-robot collaborative site inspection problem can be posed as an MMKP.

In an MMKP, we are given N_C classes of items, where each class contains J_i items, $\forall i \in \{1, \dots, N_C\}$. It is assumed that there exist N_R types of resources and the respective budgets are given by $\mathcal{E}_b = [\mathcal{E}_{b,1}, \dots, \mathcal{E}_{b,N_R}]$. In each class i , the j^{th} item is characterized by a non-negative reward $p_{i,j} > 0$ and a non-negative weight vector $G_{i,j} = [g_{i,j}^1, \dots, g_{i,j}^{N_R}] \succeq 0$. The weight $g_{i,j}^r$ denotes the needed resources from the r^{th} resource if the j^{th} item in class i is selected, for $r \in \{1, \dots, R\}$. The objective of the MMKP is then to pick exactly one item from each class such that the total reward is maximized, while satisfying all the N_R

¹See [131] for a comprehensive overview of Knapsack problems.

resource constraints. The standard MMKP can be formally stated as:

$$\begin{aligned}
& \max_z \quad \sum_{i=1}^{N_C} \sum_{j=1}^{J_i} p_{i,j} z_{i,j} \\
& \text{s.t.} \quad \sum_{j=1}^{J_i} z_{i,j} = 1, \quad \forall i \in \{1, \dots, N_C\}, \\
& \quad \quad \sum_{i=1}^{N_C} \sum_{j=1}^{J_i} g_{i,j}^k z_{i,j} \leq \mathcal{E}_{b,k}, \quad \forall k \in \{1, \dots, N_R\}, \\
& \quad \quad z_{i,j} \in \{0, 1\}, \quad \forall i \in \{1, \dots, N_C\} \text{ and } j \in \{1, \dots, J_i\},
\end{aligned} \tag{A.1}$$

where $z = [z_{1,1}, \dots, z_{1,J_1}, \dots, z_{i,j}, \dots, z_{N_C,1}, \dots, z_{N_C,J_{N_C}}]$, for $i \in \{1, \dots, N_C\}$ and $j \in \{1, \dots, J_i\}$, $z_{i,j} = 1$ indicates that item j of class i is selected and $z_{i,j} = 0$ indicates otherwise. z is then the stacked vector with all the decision variables for all the sites. When there is only one resource constraint ($N_R = 1$), this special case is referred to as the Multiple-Choice Knapsack Problem (MCKP) [117]. MMKP is then the more general case where choosing an item can cost a number of different resources.

MMKPs have found many uses in practical applications related to resource management, e.g., adaptive multimedia systems [132] and cellular network management [133], as well as in robotic applications [134–138]. However, they are typically solved either by using integer program solvers, which can be computationally expensive for large problems, or by heuristics that have no theoretical optimality guarantees.

Appendix B

B.1 Constructed Image Classification Dataset

In this dataset, there is a total of 39 object classes consisting of a variety of commonly-seen objects, and a total of 76,505 images, which are collected from COCO detection dataset [76] and ImageNet [120]. Most of the images are obtained from the COCO detection dataset by extracting object image patches based on the bounding box annotations, in order to better represent what the robot would see in real-world visual tasks. The complete list of object classes, and the numbers of images in the training, validation, and test sets are given in Table B.1.

B.2 Descriptions of Existing Methods Included in the Performance Comparisons

We provide detailed descriptions for the existing methods included in the performance comparisons in Sec. 5.3.

Joing Labeling

Independent: This is a benchmark method that directly uses the trained DCNN base classifier’s output, without considering any correlation.

Object class	Training	Validation	Test
person	1000	500	500
bicycle	1000	500	500
car	1000	500	500
motorcycle	905	400	300
airplane	1000	500	500
bus	1000	500	300
train	1000	500	500
truck	1000	500	500
boat	1000	500	500
bench	1000	500	500
bird	1000	500	500
cat	1000	500	500
dog	1000	500	500
horse	1000	500	500
sheep	1000	500	500
cow	1000	500	500
elephant	1000	500	500
zebra	1000	500	500
giraffe	1000	500	500
backpack	950	500	400
umbrella	700	450	300
suitcase	1000	500	300
bottle	1000	500	500
cup	1000	500	500
banana	1000	500	500
apple	1000	500	500
sandwich	1000	500	500
orange	1000	500	500
broccoli	1000	500	500
carrot	1000	500	500
pizza	1000	500	500
donut	1000	500	500
cake	1000	500	500
chair	1000	500	500
potted plant	1000	500	500
laptop	1000	500	500
book	1000	500	500
clock	1000	500	500
teddy bear	1000	500	500
total	38555	19350	18600

Table B.1: List of object classes in our dataset, along with the numbers of images in the training/validation/test sets.

Cao et al. [64]: This method learns image similarity using hand-crafted features (e.g., SIFT) via a Bayesian approach, after which a propagation algorithm jointly labels all the

nodes. Since their algorithm requires an initial set of correct labels, we provide it with 20% of the ground-truth labels in the comparison of Table 5.2.

Hayder et al. [65]: This method uses a Conditional Random Field (CRF) to jointly label nodes, where the edge potential is given by a similarity measure learned from data using hand-crafted features (e.g., Local Binary Pattern) and the node potential is based on the classifier output. To compare this method on our test set, we train their similarity measure on our dataset for the edge potential of their CRF and use our DCNN base classifier’s output for the node potential.

Query Selection

Independent: This is a benchmark method that selects the nodes greedily based on their respective individual uncertainty (based on the base classifier’s output).

Wang et al. [70]: Given an undirected probabilistic graphical model (e.g., MRF), this approach selects the nodes to query such that a lower bound of the expected label estimation error of the remaining nodes is minimized.

B.3 Region-Of-Interest (ROI) Extraction Algorithm

In the robotic experiments of Sec. 5.6, we used a simple saliency and depth-based algorithm to find coarse ROIs from the captured images, each of which may contain an object-of-interest. The algorithm is summarized in Alg. 5.

B.4 Background Rejection Classifier

Given an image patch from the ROI extraction algorithm, in order to determine whether this image patch does contain an object-of-interest (e.g., an object that belongs to one of the 39 classes vs. a background wall), we train a binary classifier using the

Algorithm 5: Region-of-interest extraction

INPUT: Image I and its depth map I_D with N_w and N_h being its width and height (in number of pixels), an upper bound for depth values d_{\max} .

STEP 1: Compute a saliency map I_S for I , using the method of [126].

STEP 2: Let $P_d = \{(i, j) \mid I_D(i, j) \in (0, d_{\max})\}$, which is the set of pixel indices with valid depth values.

STEP 3: Let $P_s = \{(i, j) \mid S_{i,j} \geq S_{\text{th}}, (i, j) \in P_d\}$, where S_{th} is a saliency threshold. In our implementation, we set it to be the 75th percentile of the saliency values of the pixels in P_d .

STEP 4: Let I_{ROI} be a ROI indicator map for I . We set $I_{ROI}(i, j) = 1$ if $(i, j) \in P_s$ and 0 otherwise. I_{ROI} is then dilated with a 10×10 kernel.

STEP 5: Find all the connected components (of positive-valued pixels) in I_{ROI} . A tight bounding box is drawn around each connected component, and then expanded by increasing its height and width in proportion to the dimensions of the enclosed connected component. The box's height and width are then further expanded by 60 pixels.

STEP 6: Among all the boxes, two boxes are merged under any of the following two conditions: 1) the ratio between their intersected area over the union of their areas is above 0.2, or 2) the ratio between their intersected area over the minimum of their areas is above 0.9. The boxes after the merging are the final ROIs.

AlexNet architecture.

We build the training set for this binary classifier as follows. For objects-of-interest, we use all the training images from our large image classification dataset. For background image patches, we randomly sample image patches that do not overlap with any objects-of-interest from the COCO dataset. A validation set is built similarly. The trained binary classifier has an accuracy of 0.950 on the validation set for distinguishing between object-of-interest images and background images.

Bibliography

- [1] M. Malmirchegini and Y. Mostofi, *On the spatial predictability of communication channels*, *IEEE Transactions on Wireless Communications* **11** (2012), no. 3 964–978.
- [2] Y. Mei, Y. Lu, Y. Hu, and C. Lee, *Deployment of mobile robots with energy and timing constraints*, *IEEE Transactions on Robotics* **22** (2006), no. 3 507–522.
- [3] W. M. Smith, *Urban propagation modeling for wireless systems*. PhD thesis, Stanford University, 2004.
- [4] D. Engel and C. Curio, *Pedestrian detectability: Predicting human perception performance with machine vision*, in *IEEE Intelligent Vehicles Symposium*, pp. 429–435, 2011.
- [5] M. Wakayama, D. Deguchi, K. Doman, I. Ide, H. Murase, and Y. Tamatsu, *Estimation of the human performance for pedestrian detectability based on visual search and motion features*, in *Proceedings of the International Conference on Pattern Recognition*, pp. 1940–1943, 2012.
- [6] M. P. Eckstein, C. K. Abbey, and F. O. Bochud, *A practical guide to model observers for visual detection in synthetic and natural noisy images*, *Handbook of Medical Imaging* **1** (2000) 593–628.
- [7] H. Cai and Y. Mostofi, *To ask or not to ask: A foundation for the optimization of human-robot collaborations*, in *Proceedings of the American Control Conference*, 2015.
- [8] H. Cai and Y. Mostofi, *Asking for help with the right question by predicting human performance*, in *Proceedings of Robotics: Science and Systems*, 2016.
- [9] A. M. Howard, *Role allocation in human-robot interaction schemes for mission scenario execution*, in *Proceedings IEEE International Conference on Robotics and Automation*, pp. 3588–3594, 2006.
- [10] C. Huang, M. Cakmak, and B. Mutlu, *Adaptive coordination strategies for human-robot handovers*, in *Proceedings of Robotics: Science and Systems*, 2015.

- [11] H. Koppula and A. Saxena, *Anticipating human activities using object affordances for reactive robotic response*, in *Proceedings of Robotics: Science and Systems*, (Berlin, Germany), June, 2013.
- [12] E. Sklar, S. L. Epstein, S. Parsons, A. T. Ozgelen, J. P. Munoz, and J. Gonzalez, *A framework in which robots and humans help each other.*, in *Proceedings of the AAAI Spring Symposium: Help Me Help You: Bridging the Gaps in Human-Agent Collaboration*, 2011.
- [13] V. Srivastava, *Stochastic search and surveillance strategies for mixed human-robot teams*. PhD thesis, University of California, Santa Barbara, 2012.
- [14] M. B. Dias, B. Kannan, B. Browning, E. Jones, B. Argall, M. F. Dias, M. Zinck, M. Veloso, and A. Stentz, *Sliding autonomy for peer-to-peer human-robot teams*, in *Proceedings of the International Conference on Intelligent Autonomous Systems*, pp. 332–341, 2008.
- [15] Ç. Meriçli, M. Veloso, and H. Akin, *Task refinement for autonomous robots using complementary corrective human feedback*, *International Journal of Advanced Robotic Systems* **8** (2011), no. 2 68.
- [16] R. Toris, H. Suay, and S. Chernova, *A practical comparison of three robot learning from demonstration algorithms*, in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, pp. 261–262, 2012.
- [17] R. Toris, D. Kent, and S. Chernova, *Unsupervised learning of multi-hypothesized pick-and-place task templates via crowdsourcing*, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4504–4510, 2015.
- [18] R. Kurnia, M. A. Hossain, A. Nakamura, and Y. Kuno, *Object recognition through human-robot interaction by speech*, in *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication*, pp. 619–624, 2004.
- [19] P. Rouanet, P. Oudeyer, F. Danieau, and D. Filliat, *The impact of human-robot interfaces on the learning of visual objects*, *IEEE Transactions on Robotics* **29** (2013), no. 2 525–541.
- [20] A. Sorokin, D. Berenson, S. S. Srinivasa, and M. Hebert, *People helping robots helping people: Crowdsourcing for grasping novel objects*, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2117–2122, 2010.
- [21] M. Yoshizaki, A. Nakamura, and Y. Kuno, *Mutual assistance between speech and vision for human-robot interface*, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1308–1313, 2002.

- [22] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie, *Visual recognition with humans in the loop*, in *Proceedings of the European Conference on Computer Vision*, pp. 438–451. 2010.
- [23] Y. Chen, H. Shioi, C. Montesinos, L. Koh, S. Wich, and A. Krause, *Active detection via adaptive submodularity*, in *Proceedings of the International Conference on Machine Learning*, pp. 55–63, 2014.
- [24] O. Russakovsky, L. Li, and L. Fei-Fei, *Best of both worlds: Human-machine collaboration for object annotation*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.
- [25] J. Zou and G. Nagy, *Human-computer interaction for complex pattern recognition problems*, in *Data Complexity in Pattern Recognition*, pp. 271–286. 2006.
- [26] J. A. Ratches, R. H. Vollmerhausen, and R. G. Driggers, *Target acquisition performance modeling of infrared imaging systems: past, present, and future*, *IEEE Sensors Journal* **1** (2001), no. 1 31–40.
- [27] M. P. Eckstein and J. S. Whiting, *Visual signal detection in structured backgrounds I. Effect of number of possible spatial locations and signal contrast*, *JOSA A* **13** (1996), no. 9 1777–1787.
- [28] M. P. Eckstein, A. J. Ahumada, and A. B. Watson, *Visual signal detection in structured backgrounds II. Effects of contrast gain control, background variations, and white noise*, *JOSA A* **14** (1997), no. 9 2406–2419.
- [29] J. Capó-Aponte, L. Temme, H. Task, A. Pinkus, M. Kalich, A. Pantle, and C. Rash, *Visual perception and cognitive performance*, *Helmet-Mounted Displays: Sensation, Perception and Cognitive Issues* (2009) 335–390.
- [30] “Amazon Mechanical Turk.” <https://www.mturk.com/mturk/welcome>.
- [31] A. Bechar, J. Meyer, and Y. Edan, *An objective function to evaluate performance of human–robot collaboration in target recognition tasks*, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **39** (2009), no. 6 611–620.
- [32] D. Gurari, S. Jain, M. Betke, and K. Grauman, *Pull the plug? Predicting if computers or humans should segment images*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [33] A. Mourikis and S. Roumeliotis, *Optimal sensor scheduling for resource-constrained localization of mobile robot formations*, *IEEE Transactions on Robotics* **22** (2006), no. 5 917–931.

- [34] S. He, J. Chen, Y. Sun, D. Yau, and N. K. Yip, *On optimal information capture by energy-constrained mobile sensors*, *IEEE Transactions on Vehicular Technology* **59** (2010), no. 5 2472–2484.
- [35] T.-K. Chang and A. Mehta, *Optimal scheduling for resource-constrained multirobot cooperative localization*, *IEEE Robotics and Automation Letters* **3** (2018), no. 3 1552–1559.
- [36] A. Kwok and S. Martinez, *Deployment algorithms for a power-constrained mobile sensor network*, *International Journal of Robust and Nonlinear Control* **20** (2010), no. 7 745–763.
- [37] T. Setter and M. B. Egerstedt, *Energy-constrained coordination of multi-robot teams*, *IEEE Transactions on Control Systems Technology* **25** (2017), no. 4 1257–1263.
- [38] P. S. Dutta, C. V. Goldman, and N. R. Jennings, *Communicating effectively in resource-constrained multi-agent systems.*, in *Proceedings of the International Joint Conferences on Artificial Intelligence*, 2007.
- [39] X. Ge, Q.-L. Han, and F. Yang, *Event-based set-membership leader-following consensus of networked multi-agent systems subject to limited communication resources and unknown-but-bounded noise*, *IEEE Transactions on Industrial Electronics* **64** (2017), no. 6 5045–5054.
- [40] M. Guo and M. M. Zavlanos, *Multi-robot data gathering under buffer constraints and intermittent communication*, *IEEE Transactions on Robotics* (2018).
- [41] A. Ghaffarkhah and Y. Mostofi, *Communication-aware motion planning in mobile networks*, *IEEE Transactions on Automatic Control* **56** (October, 2011) 2478–2485.
- [42] A. Ghaffarkhah and Y. Mostofi, *Dynamic networked coverage of time-varying environments in the presence of fading communication channels*, *ACM Transactions on Sensor Networks* **10** (2014), no. 3.
- [43] Y. Yan and Y. Mostofi, *To Go or Not to Go On Energy-Aware and Communication-Aware Robotic Operation*, *IEEE Transactions on Control of Network Systems* **1** (July, 2014) 218 – 231.
- [44] U. Ali, H. Cai, Y. Mostofi, and Y. Wardi, *Motion-communication co-optimization with cooperative load transfer in mobile robotics: An optimal control perspective*, *IEEE Transactions on Control of Network Systems* **6** (2018), no. 2 621–632.

- [45] M. Moser, D. P. Jokanovic, and N. Shiratori, *An algorithm for the multidimensional multiple-choice knapsack problem*, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* **80** (1997), no. 3 582–589.
- [46] S. Ren, K. He, R. Girshick, and J. Sun, *Faster R-CNN: Towards real-time object detection with region proposal networks*, in *Advances in Neural Information Processing Systems*, 2015.
- [47] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, *Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018).
- [48] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, in *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.
- [49] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, *Rethinking the inception architecture for computer vision*, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [50] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, *MobileNetV2: Inverted residuals and linear bottlenecks*, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [51] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [52] C. Galleguillos, A. Rabinovich, and S. Belongie, *Object categorization using co-occurrence, location and appearance*, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [53] A. Torralba, K. Murphy, W. T. Freeman, and M. A. Rubin, *Context-based vision system for place and object recognition*, in *International Conference on Computer Vision*, 2003.
- [54] M. Hasan and A. K. Roy-Chowdhury, *Context aware active learning of activity recognition models*, in *IEEE International Conference on Computer Vision*, 2015.
- [55] A. Jain, A. Zamir, S. Savarese, and A. Saxena, *Structural-RNN: Deep learning on spatio-temporal graphs*, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [56] E. Ahmed, M. Jones, and T. K. Marks, *An improved deep learning architecture for person re-identification*, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

- [57] S. Bell and K. Bala, *Learning visual similarity for product design with convolutional neural networks*, *ACM Transactions on Graphics* **34** (2015), no. 4 98.
- [58] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, *Deep learning for content-based image retrieval: A comprehensive study*, in *ACM International Conference on Multimedia*, 2014.
- [59] S. Zagoruyko and N. Komodakis, *Learning to compare image patches via convolutional neural networks*, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.
- [60] E. Prashnani, H. Cai, Y. Mostofi, and P. Sen, *PieAPP: Perceptual image-error assessment through pairwise preference*, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [61] J. Guérin, O. Gibaru, S. Thiery, and E. Nyiri, *CNN features are also great at unsupervised classification*, *arXiv:1707.01700* (2017).
- [62] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, *Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free*, *Robotics: Science and Systems* (2015).
- [63] S. Y. Bao, Y. Xiang, and S. Savarese, *Object co-detection*, in *European Conference on Computer Vision*, 2012.
- [64] L. Cao, J. Luo, and T. S. Huang, *Annotating photo collections by label propagation according to multiple similarity cues*, in *ACM International Conference on Multimedia*, 2008.
- [65] Z. Hayder, M. Salzmann, and X. He, *Object co-detection via efficient inference in a fully-connected CRF*, in *European Conference on Computer Vision*, 2014.
- [66] A. Anand, H. S. Koppula, T. Joachims, and A. Saxena, *Contextually guided semantic labeling and search for three-dimensional point clouds*, *The International Journal of Robotics Research* **32** (2013), no. 1 19–34.
- [67] H. Ali, F. Shafait, E. Giannakidou, A. Vakali, N. Figueroa, T. Varvadoukas, and N. Mavridis, *Contextual object category recognition for RGB-D scene labeling*, *Robotics and Autonomous Systems* **62** (2014), no. 2 241–256.
- [68] J. Ruiz-Sarmiento, C. Galindo, and J. González-Jiménez, *Joint categorization of objects and rooms for mobile robots*, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.

- [69] A. Krause, A. Singh, and C. Guestrin, *Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies*, *Journal of Machine Learning Research* **9** (2008) 235–284.
- [70] D. Wang, J. Fisher III, and Q. Liu, *Efficient observation selection in probabilistic graphical models using Bayesian lower bounds*, in *Conference on Uncertainty in Artificial Intelligence*, 2016.
- [71] B. Settles, *Active learning*, *Synthesis Lectures on Artificial Intelligence and Machine Learning* (2012).
- [72] L. Yi, V. G. Kim, D. Ceylan, I. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas, *A scalable active framework for region annotation in 3D shape collections*, *ACM Transactions on Graphics* **35** (2016), no. 6 210.
- [73] S. Chen, Y. Li, and N. M. Kwok, *Active vision in robotic systems: A survey of recent developments*, *International Journal of Robotics Research* **30** (2011), no. 11 1343–1377.
- [74] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, *Revisiting active perception*, *Autonomous Robots* **42** (2018), no. 2 177–196.
- [75] T. Patten, W. Martens, and R. Fitch, *Monte Carlo planning for active object classification*, *Autonomous Robots* **42** (2018), no. 2 391–421.
- [76] T. L. et al., *Microsoft COCO: Common objects in context*, in *European Conference on Computer Vision*. 2014.
- [77] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu, *Vision meets drones: A challenge*, *arXiv:1804.07437* (2018).
- [78] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, *Indoor segmentation and support inference from RGBD images*, in *European Conference on Computer Vision*, 2012.
- [79] R. Olfati-Saber, A. Fax, and R. M. Murray, *Consensus and cooperation in networked multi-agent systems*, *Proceedings of the IEEE* **95** (2007), no. 1 215–233.
- [80] F. Bullo, J. Cortes, and S. Martinez, *Distributed control of robotic networks: A mathematical approach to motion coordination algorithms*. Princeton University Press, 2009.
- [81] S. Timotheou and G. Loukas, *Autonomous networked robots for the establishment of wireless communication in uncertain emergency response scenarios*, in *Proceedings of the ACM symposium on Applied Computing*, 2009.

- [82] C. C. Ooi and C. Schindelbauer, *Minimal energy path planning for wireless robots*, *Mobile Networks and Applications* (2009).
- [83] F. El-Moukaddem, E. Torng, and G. Xing, *Mobile relay configuration in data-intensive wireless sensor networks*, *IEEE Transactions on Mobile Computing* **12** (2011), no. 2 261–273.
- [84] H. Jaleel, Y. Wardi, and M. B. Egerstedt, *Minimizing mobility and communication energy in robotic networks: An optimal control approach*, in *Proceedings of the American Control Conference*, 2014.
- [85] D. B. Licea, V. S. Varma, S. Lasaulce, J. Daafouz, and M. Ghogho, *Trajectory planning for energy-efficient vehicles with communications constraints*, in *Proceedings of the International Conference on Wireless Networks and Mobile Communications*, 2016.
- [86] Y. Zeng and R. Zhang, *Energy-efficient UAV communication with trajectory optimization*, *IEEE Transactions on Wireless Communications* **16** (2017), no. 6 3747–3760.
- [87] M. Lindhé and K. H. Johansson, *Using robot mobility to exploit multipath fading*, *IEEE Wireless Communications* **16** (2009), no. 1 30–37.
- [88] A. Ghaffarkhah and Y. Mostofi, *Channel learning and communication-aware motion planning in mobile networks*, in *Proceedings of the American Control Conference*, 2010.
- [89] C. Wietfeld, K. Daniel, K. P. Valavanis, and G. J. Vachtsevanos, *Cognitive networking for UAV swarms*, *Handbook of Unmanned Aerial Vehicles* (2014) 749–780.
- [90] S. Gil, S. Kumar, D. Katabi, and D. Rus, *Adaptive communication in multi-robot systems using directionality of signal strength*, *The International Journal of Robotics Research* **34** (2015), no. 7 946–968.
- [91] Y. Yan and Y. Mostofi, *Robotic router formation in realistic communication environments*, *IEEE Transactions on Robotics* (2012).
- [92] J. Fink, A. Ribeiro, and V. Kumar, *Robust control of mobility and communications in autonomous robot teams*, *IEEE Access* (2013).
- [93] Y. Wu, B. Zhang, X. Yi, and Y. Tang, *Communication-motion planning for wireless relay-assisted multi-robot system*, *IEEE Wireless Communications Letters* **5** (2016), no. 6 568–571.
- [94] A. Zhou et al., *Robotic millimeter-wave wireless networks*, *IEEE/ACM Transactions on Networking* (2020).

- [95] A. Muralidharan and Y. Mostofi, *Energy optimal distributed beamforming using unmanned vehicles*, *IEEE Transactions on Control of Network Systems* **5** (2017), no. 4 1529–1540.
- [96] D. S. Kalogerias and A. P. Petropulu, *Spatially controlled relay beamforming*, *IEEE Transactions on Signal Processing* **66** (2018), no. 24 6418–6433.
- [97] G. Sun et al., *Improving performance of distributed collaborative beamforming in mobile wireless sensor networks: A multi-objective optimization method*, *IEEE Internet of Things Journal* (2020).
- [98] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, *Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage*, *IEEE Communications Letters* **20** (2016), no. 8 1647–1650.
- [99] H. Wu, X. Tao, N. Zhang, and X. Shen, *Cooperative UAV cluster-assisted terrestrial cellular networks for ubiquitous coverage*, *IEEE Journal on Selected Areas in Communications* **36** (2018), no. 9 2045–2058.
- [100] L. Zhu et al., *3-D beamforming for flexible coverage in millimeter-wave UAV communications*, *IEEE Wireless Communications Letters* **8** (2019), no. 3 837–840.
- [101] A. Muralidharan and Y. Mostofi, *Path planning for minimizing the expected cost until success*, *IEEE Transactions on Robotics* **35** (2019), no. 2 466–481.
- [102] A. Muralidharan and Y. Mostofi, *Statistics of the distance traveled until successful connectivity for unmanned vehicles*, *Autonomous Robots* **44** (2020), no. 1 25–42.
- [103] G. A. Hollinger et al., *Underwater data collection using robotic sensor networks*, *IEEE Journal on Selected Areas in Communications* **30** (2012), no. 5 899–911.
- [104] J. U. Khan and H.-S. Cho, *A distributed data-gathering protocol using AUV in underwater sensor networks*, *Sensors* **15** (2015), no. 8 19331–19350.
- [105] S. Wang, M. Xia, and Y.-C. Wu, *Backscatter data collection with unmanned ground vehicle: Mobility management and power allocation*, *IEEE Transactions on Wireless Communications* **18** (2019), no. 4 2314–2328.
- [106] H. Cai and Y. Mostofi, *Human–robot collaborative site inspection under resource constraints*, *IEEE Transactions on Robotics* (2019).
- [107] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.
- [108] “NOAA Natural Hazards Image Database.”
<http://www.ngdc.noaa.gov/hazardimages/>.

- [109] J. Xiao, J. Hays, K. Ehinger, A. Oliva, A. Torralba, *et. al.*, *Sun database: Large-scale scene recognition from abbey to zoo*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3485–3492, 2010.
- [110] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, *The PASCAL visual object classes (VOC) challenge*, *International Journal of Computer Vision* **88** (2010), no. 2 303–338.
- [111] N. V. Chawla, *Data mining for imbalanced datasets: An overview*, in *Data Mining and Knowledge Discovery Handbook*, pp. 853–867. 2005.
- [112] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, *Caffe: Convolutional architecture for fast feature embedding*, *arXiv preprint arXiv:1408.5093* (2014).
- [113] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et. al.*, *Imagenet large scale visual recognition challenge*, *International Journal of Computer Vision* **115** (2015), no. 3 211–252.
- [114] “Pioneer 3-AT Robot.” <http://www.mobilerobots.com/ResearchRobots/P3AT.aspx>.
- [115] S. Burer and A. N. Letchford, *Non-convex mixed-integer nonlinear programming: A survey*, *Surveys in Operations Research and Management Science* **17** (2012), no. 2 97–106.
- [116] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2009.
- [117] D. Pisinger, *A minimal algorithm for the multiple-choice knapsack problem*, *European Journal of Operational Research* **83** (1995), no. 2 394–410.
- [118] A. Gonzalez-Ruiz, A. Ghaffarkhah, and Y. Mostofi, *A comprehensive overview and characterization of wireless channels for networked robotic and control systems*, *Journal of Robotics* (2011).
- [119] H. Cai and Y. Mostofi, *Exploiting object similarity for robotic visual recognition*, *IEEE Transactions on Robotics* (2020).
- [120] “ImageNet.” <http://image-net.org/explore>.
- [121] C. Bishop, *Pattern recognition and machine learning*. Springer, 2006.
- [122] D. Koller and N. Friedman, *Probabilistic graphical models: Principles and techniques*. MIT Press, 2009.

- [123] “UGM: A Matlab toolbox for probabilistic undirected graphical models.”
<http://www.cs.ubc.ca/~schmidtm/Software/UGM.html>.
- [124] A. Krause and C. Guestrin, *Optimal value of information in graphical models*, *Journal of Artificial Intelligence Research* **35** (2009) 557–591.
- [125] G. Gutin and A. P. Punnen, *The Traveling Salesman Problem and its variations*, vol. 12. Springer Science & Business Media, 2006.
- [126] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, *Salient object detection: A discriminative regional feature integration approach*, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
- [127] M. L. Puterman, *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, 1994.
- [128] C. Browne et al., *A survey of Monte Carlo tree search methods*, *IEEE Transactions on Computational Intelligence and AI in Games* (2012).
- [129] L. Kocsis and C. Szepesvári, *Bandit based Monte-Carlo planning*, in *Proceedings of the European Conference on Machine Learning*, 2006.
- [130] J. Faigl, P. Váňa, and J. Deckerová, *Fast heuristics for the 3D multi-goal path planning based on the generalized traveling salesman problem with neighborhoods*, *IEEE Robotics and Automation Letters* (2019).
- [131] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack problems*. Springer Berlin, 2003.
- [132] M. S. Khan, *Quality adaptation in a multisession multimedia system: Model, algorithms, and architecture*. PhD thesis, University of Victoria, 1999.
- [133] K. Navaie and H. Yanikomeroglu, *Optimal downlink resource allocation for non-real time traffic in cellular CDMA/TDMA networks*, *IEEE Communications Letters* **10** (2006), no. 4 278–280.
- [134] J. Bellingham, M. Tillerson, A. Richards, and J. P. How, *Multi-task allocation and path planning for cooperating UAVs*, in *Cooperative control: models, applications and algorithms*. Springer, 2003.
- [135] M. Alighanbari and J. P. How, *Decentralized task assignment for unmanned aerial vehicles*, in *Proceedings of the IEEE Conference on Decision and Control*, 2005.
- [136] M. Alighanbari and J. P. How, *Robust decentralized task assignment for cooperative UAVs*, in *Proceedings of the AIAA Conference on Guidance, Navigation, and Control*, 2006.

- [137] H. Oh, H. S. Shin, A. Tsourdos, B. A. White, and P. Silson, *Coordinated road network search for multiple UAVs using dubins path*, in *Advances in Aerospace Guidance, Navigation and Control*. Springer, 2011.
- [138] L. R. Rodrigues, J. P. P. Gomes, and J. F. L. Alcântara, *Embedding remaining useful life predictions into a modified receding horizon task assignment algorithm to solve task allocation problems*, *Journal of Intelligent & Robotic Systems* **90** (2018), no. 1-2 133–145.